UNIVERSITY OF CALIFORNIA
RIVERSIDE

RNA-Seq Based Transcriptome Assembly: Sparsity, Bias Correction and Multiple Sample Comparison

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Wei Li

September 2012

Dissertation Committee:

Dr. Tao Jiang , Chairperson
Dr. Stefano Lonardi
Dr. Marek Chrobak
Dr. Thomas Girke

The Dissertation of Wei Li is approved:

Committee Chairperson

University of California, Riverside

Acknowledgments

The completion of this dissertation would have been impossible without help from many people.

First and foremost, I would like to thank my advisor, Dr. Tao Jiang, for his guidance and supervision during the four years of my Ph.D. He offered invaluable advice and support on almost every aspect of my study and research in UCR. He gave me the freedom in choosing a research problem I'm interested in, helped me do research and write high quality papers, Not only a great academic advisor, he is also a sincere and true friend of mine. I am always feeling appreciated and fortunate to be one of his students.

Many thanks to all committee members of my dissertation: Dr. Stefano Lonardi, Dr. Marek Chrobak, and Dr. Thomas Girke. I will be greatly appreciated by the advice they offered on the dissertation. I would also like to thank Jianxing Feng, Prof. James Borneman and Paul Ruegger for their collaboration in publishing several papers. Thanks to the support from Vivien Chan, Jianjun Yu and other bioinformatics group members during my internship in the Novartis Institutes for Biomedical Research. I'm greatly appreciated to Yiqun (Eddie) Cao for his generous help and guidance on all aspects of my life.

It is my pleasure to work with my labmates: Guanqun Shi, Bob Wang, Yang Yang, Minzhu Xie, Yu-Ting Huang, Olga Tanaseichuk, Yi-Wen Yang, Elena Strzheletska, Li Yan, Dennis Duma. Also thanks to many of my friends in UCR and in life academy: Yang Zhao, Xiaoqing Jin, Yingdi Liu, Xiaoquan Zhou, Lexiang Ye, Ning Mi, Xiaoyue Wang, Duo Li, Shan Shan, Changhui Lin, and many others. We shared many wonderful moments in our lives.

The dissertation is dedicated to my parents, Linping Li and Meixiang Liu, for their love and giving in the past 28 years. I would also thank my girlfriend, Wenting Wang; without your support, this dissertation is impossible.

ABSTRACT OF THE DISSERTATION

RNA-Seq Based Transcriptome Assembly:
Sparsity, Bias Correction and Multiple Sample Comparison

by

Wei Li

Doctor of Philosophy, Graduate Program in Computer Science
University of California, Riverside, September 2012
Dr. Tao Jiang , Chairperson

RNA-Seq, or deep-sequencing of RNAs, is a new technology for transcriptome profiling using second generation sequencing. RNA-Seq has been widely used to identify and quantify transcriptomes at an unprecedented high resolution and low cost. An important computational problem arising from RNA-Seq is transcriptome assembly, in which the structures of transcripts (and their expression levels) are inferred simultaneously from RNA-Seq data. RNA-Seq transcriptome assembly allows for the detection of structural and quantitative changes of transcripts between samples, paving the way for novel biological discoveries. However, the problem of RNA-Seq transcriptome assembly is challenging because: (i) the complicated alternative splicing patterns of some genes result in a huge number of possible transcripts, (ii) different kinds of biases in RNA-Seq reads (including sequencing, positional and mappability biases) decrease the accuracy of assembly and expression level estimation algorithms, and (iii) the existing assembly tools can only reconstruct transcripts from a single sample, leading to a high false positive rate for comparing RNA-Seq experiments from multiple samples.

We propose three different algorithms to address these challenges. First, we design a transcriptome assembly tool, IsoLasso, that balances different objectives (pre-

diction accuracy, sparsity, interpretation) and takes advantage of the sparsity of expressed transcripts. Second, we use the quasi-multinomial distribution to model the RNA-Seq biases, and design a new algorithm, CEM, to handle different biases in both transcriptome assembly and transcript expression level estimation. Finally, we propose a multiple-sample transcriptome assembly tool, ISP, to assemble transcripts directly from RNA-Seq data of multiple samples. ISP reaches an improved performance compared to the assembly tools that consider one sample at a time, and helps to improve the accuracy of downstream differential analysis of transcriptomes between samples.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 RNA-Seq and transcriptome assembly

The second generation sequencing (or high throughput sequencing, next generation sequencing) technology is a new technology to sequence DNAs at an unprecedented high data throughput and low cost. This new sequencing technology has become a revolutionary tool in many research areas, including medical sciences [33, 37], microbiology [7], genetics [32, 96, 25], evolution [5], *etc.* Transcriptomic research has taken advantage of the second generation sequencing methods, leading to a new experimental protocol, RNA-Seq [58, 97, 48, 56, 51, 8, 59]. RNA-Seq sequences the complementary DNAs (cDNAs) which are reversely transcribed from RNA sequences of the cells being studied, and the sequencing data reveals the structural and quantitative properties of the cell's transcriptomes. RNA-Seq has become a fundamental and popular protocol in transcriptomic research [27, 57, 88, 94], but it also posts many challenging problems for computational biology, including read mapping [86], normalization [71], between-sample comparison [91], transcriptome assembly [87], isoform expression level estimation [38], *etc.*

The problem of transcriptome assembly and transcript expression level estimation is to reconstruct the structures of transcripts (or isoforms), and to estimate their expression levels from RNA-Seq reads. Based on the transcriptome assembly results, one can detect new isoforms and analyze their structural and quantitative changes between samples, for example, to detect novel oncogenes and splicing variants that are associated with disease or cancer [33, 37]. Due to the importance of this problem, numerous tools are developed using either the *ab initio* approach or the *de novo* approach. The difference between both approaches is whether the reference genome is used: in the *ab initio* approach, a splicing detection software (like TopHat [86] and SpliceMap [2]) is required to map RNA-Seq reads to the reference genome. Examples of *ab inito* assembly tools include Cufflinks [87], Scripture [26], IsoInfer [18], IsoLasso [47], SLIDE [45], *etc.* In contrast, the *de novo* approach assembles transcripts without the help of the reference genome; these tools include AbySS [76], Trinity [23], Rnnotator [52], *etc.* Compared to the *de novo* approach, the *ab initio* approach usually provides better assembly results, since it uses additional information from the reference genome. It also requires less computational resources [53].

In this dissertation, we study three different problems in transcriptome assembly: how to take advantage of the sparsity in expressed isoforms, how to correct biases in read distribution along an isoform, and how to analyze RNA-Seq data from multiple samples. In the following, a short introduction to each of the three problems is presented in Section 1.2, 1.3 and 1.4, respectively.

1.2 Sparsity in transcriptome assembly

Different *ab initio* transcriptome assembly tools use different strategies. IsoInfer [18] enumerates all possible "valid" isoforms and uses a quadratic program (QP) to estimate the expression levels of a given set of isoforms. IsoInfer then chooses the best subset of valid isoforms such that the estimated abundance of every "expressed segment" of the reference genome (*e.g.*, an exon) is proportional to the observed reads falling into the segment. On the other hand, Cufflinks [87] assembles isoforms using a parsimony strategy, *i.e.*, it attempts to identify the minimum number of isoforms to cover all the reads. To do this, Cufflinks decomposes the "overlap graph" of compatible reads into a smallest path cover, and then calculates the expression levels of the isoforms (*i.e.*, paths in the cover) using the probabilistic model proposed in [38].

The strategies that IsoInfer and Cufflinks adopted correspond to two different model selection principles: *prediction accuracy* and *interpretation* (or *sparsity*) [29]. IsoInfer selects isoforms to maximize the prediction accuracy, *i.e.*, to minimize the error or discrepancy between the predicted and observed expression levels in all expressed segments. IsoInfer employs a search algorithm similar to the "best subset variable selection" algorithm [30] to find the best subset of isoforms. However, the huge search space prevents the algorithm from doing a thorough search, and many heuristic restrictions must be applied to make the search tractable. On the other hand, Cufflinks minimizes interpretation, *i.e.*, the number of variables (or isoforms) that are required to explain all the mapped reads. Here, the prediction accuracy is not considered explicitly during the transcriptome assembly process. By defining a "partial order" between reads, Cufflinks filters out "uncertain" paired-end reads which may result in a sub-optimal path cover in the solution, or miss some alternative splicing events. Finally, Scripture [26]

reconstructs all possible isoforms by enumerating all possible paths in the "connectivity graph". This approach may lead to many incorrect isoforms for complex genes with a large number of exons, since the number of paths may be huge for such gene models.

Another important objective in transcriptome assembly is *completeness*, which requires that all exons (and exon junctions) appear in at least one isoform in the solution (as done in IsoInfer [18]), or all mapped reads be contained in at least one isoform (as done in Cufflinks [87]). In IsoInfer, the completeness is achieved by solving a set cover instance that covers all expressed segments and exon junctions. Since all the reads represented in the overlap graph are partitioned into disjoint paths in Cufflinks, they are guaranteed to be supported by at least one isoform (*i.e.*, path). However, some "uncertain" paired-end reads (*i.e.*, reads that cannot be included in partial order and thus absent in the overlap graph) may not be covered by the solution. Scripture adopts a conservative approach to enumerate all possible paths in its connectivity graph, which is guaranteed to cover all expressed segments and exon junctions. Like Cufflinks, the prediction accuracy is not considered explicitly during the transcript assembly process of Scripture. Moreover, retaining all possible isoforms clearly leads to a bad interpretation. Table 1.1 lists all the principles (or objectives) that IsoInfer, IsoLasso (the algorithm to be introduced in Chapter 2), Cufflinks and Scripture abide by in the transcript assembly process.

In Chapter 2, we will present IsoLasso, a new transcriptome assembly algorithm which balances prediction accuracy, sparsity and completeness. IsoLasso uses the LASSO algorithm to balance both prediction accuracy and sparsity, and we further take completeness into our consideration by adding constraints to the quadratic programming problem originally used by LASSO. We also prove some theorems concerning the predicted isoforms of IsoLasso and Cufflinks, and show that IsoLasso is able to handle

Table 1.1: Transcriptome assembly principles (or objectives) of IsoInfer, Cufflinks, Scripture and IsoLasso. Theoretically both Cufflinks and IsoLasso take completeness into consideration, but in practice they may not fully guarantee it and thus are marked "partially" in the table.

Algorithm	Prediction accuracy	Interpretation	Completeness
IsoInfer	Yes	Partially	Yes
Cufflinks	No	Yes	Partially
Scripture	No	No	Yes
IsoLasso	Yes	Yes	Partially

uncertain paired-end reads which are discarded by Cufflinks.

1.3 Handling RNA-Seq biases in transcriptome assembly and isoform expression level estimation

RNA-Seq biases refer to the non-random, non-uniform distributions of the sequenced reads across the involved isoforms (or genes) in an RNA-Seq experiment. The bias pattern varies depending on several factors, for example, the GC content of the sequence [66], experimental protocol [70], sequencing platform [28], repeat sequences of the genome [65], the secondary structure of RNAs [46], *etc.* Both *positional* [15, 58] and *sequencing* biases [28, 46] are routinely observed in RNA-Seq experiments. Positional bias is the non-uniform distribution of reads over different positions of a transcript, while sequencing bias refers to the distribution of reads related to the sequence content and the priming method used in library preparation [46]. Since many second generation sequencing applications (including RNA-Seq) require the mapping of reads to the reference genome, the *mappability* bias is also an important source of biases in RNA-Seq [75] and ChIP-Seq [72, 6]. The mappability bias arises when read counts are biased due to the read mapping. For example, some reads may not be mapped due to sequencing errors and some applications discard reads mapped to the repeat regions of the reference

genome; the numbers of reads are thus under-counted for these regions. Also, incorrect read mapping leads to incorrect read counts for regions where the involved reads are mapped to. It is important to notice that different types of biases may be related to the same factor. For example, the GC content of the genomic sequence may affect both sequencing and mappability biases [15, 28, 6].

Biases in RNA-Seq data may cause inaccurate expression level estimations of genes (or isoforms), since many methods use a simplifying assumption that reads are uniformly sampled (called "the Poisson assumption") [38, 18, 47]. Currently, most bias correction methods try to overcome the effect of biases on gene and isoform expression level estimations. For example, positional biases are handled in [99] by learning non-uniform read distributions from given RNA-Seq data or by modeling the RNA degradation [89]. In [79], a generalized Poisson (GP) model is used to calculate the expression levels of genes affected mainly by sequencing biases. Other approaches include checking the repeat regions of the reference genome to handle mappability biases [69, 42], modeling the dependency between neighboring positions to correct sequencing biases [46], or a combination of several strategies. For example, [70] corrects both positional and sequencing biases by combining the machine learning techniques in [99] and a probabilistic generative model similar to [46]. However, drawbacks exist in these methods. On the one hand, some methods can handle only one specific type of biases (*e.g.*, [99, 69]) or correct biases only at the gene level [79]. On the other hand, more general methods such as [46] and [70] use sophisticated probabilistic models that require the learning of a large number of parameters, and thus have to make some simplifying assumptions to make the computation tractable [70].

Besides expression level estimation, the RNA-Seq biases also have significant effects on transcriptome assembly. Transcriptome assembly tools that consider predic-

tion accuracy are affected by RNA-Seq biases, if these biases lead to inaccurate isoform expression level estimations. Also, RNA-Seq biases may generate "gap regions" on the reference genome where no mapped reads are observed. Because of these gaps, two broken transcripts may be assembled instead of one complete transcript. Furthermore, incorrectly mapped reads may lead to incorrect transcript assemblies. As far as we know, most work in the literature concerning RNA-Seq biases deal with correcting gene (or isoform) expression level estimation, and the effects of biases on the transcriptome assembly remain largely unexplored.

In Chapter 3, we will introduce a new algorithm, CEM, that considers positional, sequencing and mappability biases in both transcriptome assembly and isoform expression level estimation. Based on a statistical framework using the quasi-multinomial distribution to capture biases, CEM uses a component elimination Expectation-Maximization (EM) algorithm to assemble transcripts and estimate their expression levels.

1.4 Transcriptome assembly from multiple sample RNA-Seq data

Both *ab initio* approach and *de novo* approaches of RNA-Seq transcriptome assembly have their advantages and disadvantages. Compared with *de novo* methods, *ab initio* transcriptome assembly algorithms use additional information from the reference genome, and thus are able to recover transcripts with a better accuracy and yet demand less computational resource [53]. However, current *ab initio* assembly algorithms critically depend on the quality of the reference genome and mapping software, and they are not specifically designed to handle errors in mapped reads, especially junction reads

which are the main evidences of splicing. These erroneous RNA-Seq reads may come from various sources, including unwanted RNA fragments during the library preparation and the mapping errors (due to sequencing errors and/or repeats). Also, it has been observed in practice that RNA-Seq reads contain a large number of "dark matters", many of which come from inter-genetic and intron regions. The origin of these reads is unclear, but a general conjecture is that they are experimental artifacts, or derived from intron retention or non-coding RNAs [67]. Clearly, if the reference genome contains errors (especially insertions and deletions in splice junction regions), the read mapping software may report incorrectly mapped reads or fail to report junctions reads.

In many RNA-Seq based studies, multiple sample RNA-Seq datasets are available. It is now common for an RNA-Seq project to sequence the whole transcriptomes of samples obtained from multiple replicates, tissues, populations, *etc.*, For example, the Encyclopedia Of DNA Elements (ENCODE) project [81] aims at creating functional element profiles of more than 100 human cell types. More than 200 RNA-Seq datasets from various tissues and experimental protocols are available for public use [83]. Other large research projects that are producing many multiple sample RNA-Seq data include The Cancer Genome Atlas (TCGA, [80]), the Model Organism ENCyclopedia of DNA Elements (modENCODE, [84]), *etc.* On the one hand, RNA-Seq reads from multiple samples could potentially help assemble transcripts better than using only one sample, since the samples can be correlated. On the other hand, transcriptome assembly for multiple samples and subsequent differential analysis are more challenging because (i) multiple sample RNA-Seq data typically contains more noise and (ii) differential analysis is sensitive to assembly and abundance estimation errors. Therefore, to analyze the structural and quantitative differences of isoforms from multiple samples, a highly accurate transcriptome assembly and abundance estimation tool working on multiple

sample RNA-Seq data is necessary.

A straightforward way to assemble transcriptomes for multiple samples is to assemble transcripts for each sample separately and "merge" all transcripts to obtain a "universal" set of isoforms, which is then used for downstream applications including abundance estimation and differential analysis. An example of this approach includes the "Cuffmerge","Cuffdiff" and "Cuffcompare" programs in the Cufflinks software package [87]. However, as more samples are sequenced, errors from individual assemblies are likely to accumulate, which could seriously affect the isoform abundance estimation and result in unreliable (or even misleading) differential analysis results.

In Chapter 4, we will present our new multiple sample transcriptome assembly tool, ISP, that is able to handle noisy RNA-Seq reads and multiple sample RNA-Seq datasets effectively. ISP reconstructs transcripts directly from multiple samples, and takes advantage of the extra information contained in multiple sample RNA-Seq datasets. By solving a linear programming problem iteratively, ISP achieves a high performance by discarding problematic reads and recovering missing junctions caused by various errors.

1.5 Publications

This dissertation encompasses three publications. The IsoLasso paper (including Section 1.2 and Chapter 2) is published in the 15th Annual International Conference on Research in Computational Molecular Biology (RECOMB 2011) and in Journal of Computational biology [47]. The CEM paper (including Section 1.3 and Chapter 3) is under review in Bioinformatics. The complete list of publications includes:

- Wei Li, Jianxing Feng and Tao Jiang. IsoLasso: A LASSO Regression Approach

to RNA-Seq Based Transcriptome Assembly. 15th Annual International Conference on Research in Computational Molecular Biology (RECOMB 2011), Lecture Notes in Computer Science, 6577:167-188, Springer Berlin/Heidelberg, 2011. Also appears in Journal of Computational Biology, 18(11): 1693-1707, Nov. 2011.

- Wei Li and Tao Jiang. Transcriptome Assembly and Isoform Expression Level Estimation from Biased RNA-Seq Reads. Submitted to Bioinformatics.

- Wei Li and Tao Jiang. Accurate Isoform Inference and Abundance Estimation from Multiple Sample RNA-Seq Data. In preparation.

Chapter 2

A LASSO Regression Approach to RNA-Seq Based Transcriptome Assembly

2.1 Introduction

In this chapter, we present a new isoform assembly algorithm, IsoLasso, which balances prediction accuracy, interpretation and completeness. IsoLasso uses the LASSO algorithm, or Least Absolute Shrinkage and Selection Operator [85], which is a shrinkage least squares method in statistical machine learning. By adding an L1 norm penalty term to the least squares objective function, LASSO achieves sparsity by setting the expression levels of unrelated isoforms to zero, thus balancing both prediction accuracy and interpretation. The LASSO algorithm is widely applied in many computational biology areas, such as genome-wide association analysis [98, 39], gene regulatory network [24], microarray data analysis [49], *etc.* In IsoLasso, we expand the quadratic

programming problem in LASSO to take completeness into consideration. Our experiments demonstrate that IsoLasso runs efficiently and achieves overall higher sensitivity and precision than IsoInfer, Cufflinks and Scripture.

The rest of Chapter 2 is organized as follows. Sections 2.2.1 and 2.2.2 present our algorithm for generating (or enumerating) candidate isoforms and its relationship to minimum path covers used in Cufflinks [87]. These candidate isoforms will be fed to our LASSO algorithm described in Section 2.2.3 for estimating isoform expression levels (or, equivalently, for inferring expressed isoforms). Section 2.2.4 expands the basic LASSO approach to take completeness into consideration. Experimental results are presented in Section 2.3, which include comparisons between IsoLasso, IsoInfer, Cufflinks, and Scripture on simulated and real datasets. Section 2.4 concludes this chapter.

2.2 Methods

2.2.1 Enumerating candidate isoforms

IsoInfer [18], Scripture [26] and Cufflinks [87] enumerate candidate isoforms in different ways. IsoInfer, assuming that expressed segment (or exon) boundaries in a gene are given, enumerates all possible combinations of segments. Note that it is possible that some lowly expressed segment are not hit by short reads and thus many of the isoforms enumerated by IsoInfer might have very low expression levels. Scripture enumerates all possible maximal paths in a *connectivity graph*; but some of these isoforms may be "infeasible" because they cannot be assembled from the mapped reads (Figure 2.2 shows such an example). Cufflinks tries to build an *overlap graph* from partially ordered reads, and assembles putative transcripts by decomposing the overlap graph into a parsimonious path cover. However, a strict partial order between reads is required here.

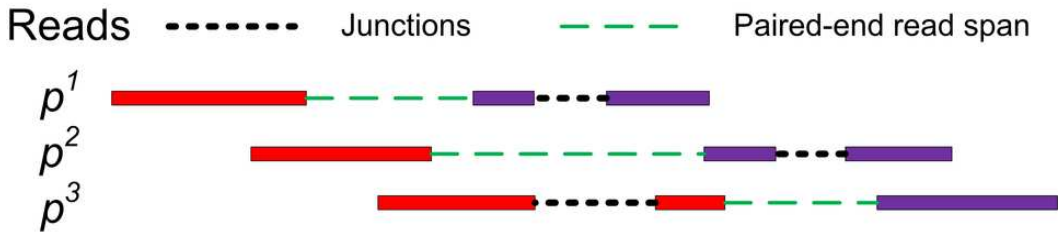


Figure 2.1: Removal of "uncertain" reads may cause splicing junctions undetected in Cufflinks. Three paired-end reads, p^1, p^2 and p^3, concern different splicing junctions. Both pairs (p^1, p^2) and (p^2, p^3) are compatible, but the pair (p^1, p^3) is not. Removing any of these reads will cause one or more junctions undetected.

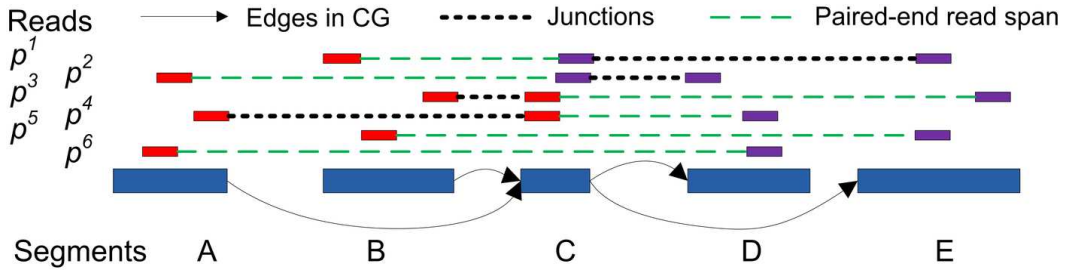


Figure 2.2: "Infeasible" paths in the connectivity graph. In the example above, there are four possible combinations of segments: ACD, ACE, BCD, and BCE. However, ACE and BCD are infeasible since they cannot be assembled from the mapped paired-end reads.

Since the actual sequence between the ends of each paired-end read is unknown, Cufflinks has to exclude some paired-end reads (called *uncertain reads*) to maintain the partial order. Removing uncertain reads may lead to two potential problems: (1) the path cover solution is actually sub-optimal and (2) some alternative splicing events are missed, if the reads including these events are removed. For instance, Figure 2.1 provides an example that removing such "uncertain" reads leaves some splicing junctions undetected. Note that uncertain reads should be treated separately from repeat sequences or incorrectly mapped reads.

Here, we describe our method of enumerating isoforms based on the connectivity graph [26] in Algorithm 1, from which the enumerated isoforms will be the set of

candidate isoforms to be considered in the LASSO algorithm. The algorithm first enumerates isoforms from the connectivity graph as in [26], and then uses two additional steps to remove isoforms that are impossible to assemble. We will prove some important properties of Algorithm 1: if there are no "uncertain" reads, then every isoform output by Algorithm 1 can be assembled from a maximal path in the overlap graph given in [87]. Moreover, the isoforms enumerated by Algorithm 1 form a superset of all possible maximal paths in the overlap graph. In other words, our LASSO algorithm in general considers more isoforms than Cufflinks in the transcript assembly process. Before giving a detailed description of this algorithm and proofs of these properties, we first briefly review some necessary notations first introduced in [87] and [26].

Algorithm 1 Isoform Enumeration

Require: A CG $G = (V, E)$, and a set of mapped single-end or paired-end reads R

Enumeration:
T (the set of isoforms) $\leftarrow \emptyset$
for $v_j \in V$ with $indeg(v_j) = 0$ **do**
  Enumerate all possible maximal paths P that begin at v_j and end at some v_k with $outdeg(v_k) = 0$
  $T \leftarrow T \cup P$
end for
Filtration:
for $t \in T$ **do**
  Let $t' = OR(\{b \in R | b \sim t\})$
  $T \leftarrow (T \backslash \{t\}) \cup \{t'\}$
end for
Condensation:
for $t \in T$ **do**
  Let $R_t = \{b \in R|, b \sim t\}$
  for $t' \in T \backslash \{t\}$ **do**
    Let $R_{t'} = \{b \in R|, b \sim t'\}$
    if $R_t \subset R_{t'}$ **then**
      $T \leftarrow (T \backslash \{t\})$
    end if
  end for
end for
Output T

2.2.1.1 Isoforms, single-end and paired-end mapped reads

A gene sequence S of length n is an ordered character sequence $S = S_1 S_2 \cdots S_n$, where $S_i \in \{A, T, G, C\}$. Define $B(n)$ as the set of binary vectors of length n. For a vector $b \in B(n)$, $b_i \in \{0, 1\}$ indicates the ith element of vector b. For a subset $U \subset B(n)$, define $OR(U) = b \in B(n)$ with $b_i = 1$ iff there is an element $c \in U$ such that $c_i = 1$. For a binary vector $b \in B(n)$, define the start (or end) of b as the first (or last) non-zero index of b, and is denoted as $l(b)$ (or $u(b)$). Hence, each isoform on gene S could be represented as a binary vector $b \in B(n)$ with $b_i = 1$ iff the nucleotide S_i is included in this isoform. A single-end or paired-end read mapped to S could also be represented as an element $b \in B(n)$ with $b_i = 1$ iff this read contains S_i. A paired-end read is denoted as $p = (b^1, b^2)$, where b^1 and b^2 are the two mapped single-end reads, and $l(b^1) < l(b^2)$. Given a set of single-end or paired-end reads R, the coverage of S_i, or $cvg(S_i)$, is the number of reads b with $b_i = 1$.

2.2.1.2 Compatibility and Connectivity Graph (CG)

The compatibility between a read b and an isoform t indicates whether b is possible to come from t. A single-end read b is *compatible* with t, denoted as $b \sim t$, iff $b_i = t_i$ for $l(b) \leq i \leq u(b)$. Similarly, a paired-end read $p = (b^1, b^2)$ is compatible with isoform t (denoted as $p \sim t$) iff $b^1 \sim t$ and $b^2 \sim t$. Given a set of single-end (or paired-end) reads R mapped to gene S, the *connectivity graph (CG)* [26] is a directed acyclic graph (DAG) $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ and $e = (v_i, v_j) \in E$ iff one of the following conditions is true:

Condition 1. There exists a single-end read or an end of some paired-end read $b \in R$ such that $b_i = 1$, $b_j = 1$, and $b_k = 0$, $\forall i < k < j$;

Condition 2. $cvg(S_i) > 0$, $cvg(S_j) > 0$, and $cvg(S_k) = 0$, $\forall i < k < j$.

Condition 2 is designed to connect two mapped reads separated by a coverage gap. Based on the definition of CG, a path h in the CG could be readily treated as an isoform by defining the isoform t as $t_i = 1$ iff $v_i \in h$. Therefore, a read b is compatible with h (denoted as $b \sim h$) iff $b \sim t$.

The isoform enumeration algorithm depicted in Algorithm 1 takes the CG as the input, and outputs a set of maximal candidate isoforms T. The algorithm consists of three phases, Enumeration, Filtration and Condensation. In the Enumeration phase, all maximal paths in the CG are enumerated. However, some of these isoforms are "infeasible" in the sense that they cannot be assembled from the mapped reads (see Figure 2.2 for an example). In this case, the second phase (*i.e.*, the Filtration phase) is required to remove such isoforms. For each isoform t generated in the Enumeration phase, the Filtration phase first finds all reads that are compatible with t, and then checks if t can be assembled from these compatible reads (it replaces t otherwise). Finally, the Condensation phase removes all the isoforms that are not maximal candidates.

2.2.2 A connection to Cufflinks

Cufflinks assembles transcripts based on the *overlap graph (OG)*, which is constructed from a set of mapped single-end or paired-end reads after removing *uncertain* reads and extending reads to include their *nested* reads [87]. It generates transcripts by partitioning the overlap graph into a *minimum path cover*, where a path cover is a set of disjoint paths in the overlap graph such that every read appears in one and only one path. A minimum path cover is a path cover with the minimum number of paths.

We will prove some theorems to establish the relationship between the set of isoforms generated by Algorithm 1 and the set of transcripts that could be constructed from the overlap graph. The formal definitions of uncertain reads, nested reads and the overlap graph are given in [87], and are reviewed below for the reader's convenience.

A single-end read b is *nested* in another single-end read b' iff $b_i = b'_i, l(b) \leq i \leq u(b)$, and at least one of the following two conditions is true: (1) $l(b) \neq l(b')$ and (2) $u(b) \neq u(b')$. A paired-end read p is *nested* in another paired-end read p' iff $l(p) \geq l(p')$, $u(p) \leq u(p')$ and at least one of the following conditions is true: (1) $l(p) \neq l(p')$ and (2) $u(p) \neq u(p')$. If a single-end read b is nested in b', b can always be removed safely without losing any information.

Two single-end reads b and b' are *compatible*, denoted as $b \sim b'$, iff there exists one isoform t such that $b \sim t$, $b' \sim t$, and b and b' are not *nested* to each other. If b and b' are not compatible, we denote $b \nsim b'$. Two paired-end reads p and p' are *compatible*, denoted as $p \sim p'$, iff there exists an isoform t such that $p \sim t$, $p' \sim t$ and p is not nested in p' or *vice versa*. If p and p' are not compatible, we denote $p \nsim p'$.

Define a *partial order* $\leq$ between two single-end reads b and b': $b \leq b'$ iff $b \sim b'$ and $l(b) \leq l(b')$. It is impossible to extend the partial order to paired-end reads, since the sequence within a paired-end read is not completely known. Alternatively, for two paired-end reads p and p', define $p \leq p'$ *with respect to a given read set* R iff the following conditions are true: (1) $p \sim p'$, (2) $l(p) \leq l(p')$, $u(p) \leq u(p')$, and (3) there is no paired-end read $p'' \in R$ such that $p \sim p'$, $p \sim p''$ but $p \nsim p''$. Write $p \leq p''|R$ if $p \leq p'$ with respect to a given read set R, or write simply $p \leq p'$ if there is no ambiguity. If reads p, p' and p'' exist such that $p \sim p'$, $p' \sim p''$ and $p \nsim p''$, then p, p' and p'' are said to be *uncertain* since no partial order can be given to these reads.

Given a set of mapped single-end or paired-end reads $R = \{b^1, b^2, \dots\}$, the over-

lap graph (OG) [87] is a DAG $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_{|R|}\}$ and $e = (v_i, v_j) \in E$ iff $b^i \leq b^j$. A *maximal path* of length k on the OG is a path $h = \{v_{i_1} \leq v_{i_2} \leq \cdots \leq v_{i_k}\}$ on the OG, such that there exists no path $h' = \left\{v_{j_1} \leq v_{j_2} \leq \cdots \leq v_{j_{k'}}\right\}$ with $h \subset h'$. Because the vertices in the OG have a one-to-one relationship with the mapped reads, we also treat vertices in the OG as binary vectors to simplify notations below. For example, if a path $h = \{v_{i_1} \leq v_{i_2} \leq \cdots \leq v_{i_k}\}$, we will use $OR(h)$ to denote $OR(\{b^{i_1} \leq b^{i_2} \leq \cdots \leq b^{i_k}\})$.

Consider a fixed gene S and the set of reads R mapped to S. We introduce some necessary lemmas, followed by two theorems and one corollary concerning the isoforms generated by OG and CG.

Lemma 1 *Denote the vertex set of the CG as $V = \{v_1, v_2, \ldots, v_n\}$. For $1 \leq i < j \leq n$, there is a path from v_i to v_j if $cvg(S_i) > 0$ and $cvg(S_j) > 0$.*

Proof. We use an induction on $n = j - i$ to prove this lemma. If $j - i = 1$, then there is an edge between v_i and v_j by Condition 2 of the CG's edge construction. Assume that $\forall k < n$, there is a path from v_i to v_j if $cvg(S_i) > 0$ and $cvg(S_j) > 0$, $j - i = k$. For $k = n$, if $cvg(S_l) = 0$ for every $i < l < j$, then there is an edge between v_i and v_j by Condition 2 of the CG's edge construction. Otherwise, if there exists $i < l' < j$ such that $cvg(S_{l'}) > 0$, then $l' - i < n$ and $j - l' < n$. Using the assumption above, there is a path from v_i to $v_{l'}$ and a path from $v_{l'}$ to v_j. Therefore, there is a path from v_i to v_j. ■

Lemma 2 *For any read set $Q \subseteq R$, if every two reads in Q are compatible, then there is a maximal path h in the CG such that $\forall b \in Q$, $b \sim h$.*

Proof. Let $t = OR(Q)$. We construct h by defining its vertex set $V(h)$ and edge set $E(h)$ separately. For every $1 \leq i < m, t_i = 1$, if the set $\{k > i | t_k = 1\}$ is not empty,

denote $j = min_k\{k > i, t_k = 1\}$. If there is a read $b \in Q$ such that $b_i = b_j = 1$ and $b_k = 0, i < k < j$, then there must be an edge e in CG from v_i to v_j by Condition 2 of CG's edge construction, and we put e in $E(h)$. Otherwise, there must be a path h' from v_i to v_j by Lemma 1, because $cvg(S_i) > 0$ and $cvg(S_j) > 0$. We put edges in h' in $E(h)$. Define $V(h)$ as the set of vertices induced by $E(h)$. A trivial case is that $|\{1 \leq i < m, t_i = 1\}| = 1$. In this case, let $V(h) = v_i, t_i = 1$ for completeness.

We claim that all reads in Q are compatible with h. This is because for a single-end read (or an end of some paired-end read) b in Q, if $b_i = 1$ then $v_i \in V(h)$. If $b_i = b_j = 1$ and $b_k = 0, i < k < j$, v_i and v_j are directly connected by edge (v_i, v_j) in h, which means that $\{v_k | i < k < j\} \cap V(h) = \emptyset$. Therefore $b \sim h$.

Once h is obtained, it is easily extended to a maximal path without violating its compatibility with every read in Q. ■

Lemma 3 *Suppose that R has no uncertain or nested reads. For every maximal path h on the OG constructed based on R, $OR(h) \in T$.*

Proof. Let $t = OR(h)$ and R_t be the set of reads corresponding to path h. By Lemma 2, there is a maximal path h' on the CG such that every read $b \in R_t$ is compatible with h'. Denote the isoform corresponding to h' as t'. Then, $t' \in T$ after the Enumeration phase of Algorithm 1 and $b \sim t'$.

Let $R_{t'} = \{b \in R | b \sim t'\}$. For any $b \in R_t$, $b \sim t'$ so $b \in R_{t'}$, then we have $R_t \subseteq R_{t'}$. Furthermore, for any $b' \in R_{t'}$, $b' \sim t'$, and thus we have $b \sim b', \forall b \in R_t, \forall b' \in R_{t'}$. If there is a read $b \in R_{t'}$ but $b \notin R_t$, the vertex corresponding to b in the OG could be added to path h, because b is compatible with all the reads in R_t and b is not a nested or uncertain read. However, this contradicts the assumption that h is maximal. Therefore, $R_t = R_{t'}$ and $t \in T$ after the Filtration phase of Algorithm 1. Note that t would not be

removed in the Condensation phase Algorithm 1 because t is maximal. ■

Lemma 4 *Suppose that R has no uncertain or nested reads. For every isoform t output by Algorithm 1, there exists a maximal path h on the OG such that $OR(h) = t$.*

Proof. Let t be an isoform enumerated by Algorithm 1 and $R_t = \{b \in R | b \sim t\}$. Since R contains no uncertain or nested reads, the vertices corresponding to R_t in the OG form a path h. If h is not maximal, it can be "expanded" to a maximal path h' by adding some vertices not in h. According to Lemma 3, there is an isoform $t' \in T$ such that $t' = OR(h')$. Denoting $R_{t'} = \{b \in R | b \sim t'\}$, then we have $R_t \subset R_{t'}$. Therefore, t would be removed in the Condensation phase of Algorithm 1, which contradicts the fact that t is output by Algorithm 1. ■

Lemmas 3 and 4 immediately lead to Theorem 1 and its corollary, Corollary 1, below.

Theorem 1 *Suppose that R contains no uncertain or nested reads. If we denote the set of isoforms constructed by Algorithm 1 as T and the set of the isoforms formed by enumerating maximal paths on the OG (constructed from R) as T_{OG}, then $T = T_{OG}$.*

Corollary 1 *If R contains no uncertain or nested reads, then for every minimum path cover H of the OG, there exists a set of maximal isoforms $T' = \{t^1, \ldots t^m\} \subset T$ such that $m = |H|$ and for every read b on a path $h \in H$, $b \sim t^i$, $1 \leq i \leq m$.*

Note that each nested read r in R is removed in [87] by extending the reads that r is nested in. On the other hand, if there are uncertain reads in R, Algorithm 1 may generate some isoforms that do not correspond to any paths on the OG when these uncertain reads cover some unique splicing junctions as shown in Figure 2.1. The following theorem states the relationship between maximal paths on the OG and the isoforms generated by Algorithm 1 when uncertain reads are present in R.

Theorem 2 *Suppose that no reads in R are nested and denote the set of isoforms constructed by Algorithm 1 as T. For every maximal path h on the OG constructed by removing uncertain reads in R, T contains an isoform which is compatible with every read on the path h.*

Proof. The proof is similar to the proof of Lemma 3. Let $t = OR(h)$ and $1 \leq l_1 < l_2 < \cdots < l_m \leq n$ be indices in t such that $t_i = 1$ iff and only if $i \in \{l_1, l_2, \ldots, l_m\}$. Let R_t be the set of reads corresponding to path h. By Lemma 2, there is a maximal path h' on the CG such that every read $b \in R_t$ is compatible with h'. Denote the isoform corresponding to h' as t'. Therefore, $t' \in T$ after the Enumeration phase of Algorithm 1 and $b \sim t'$.

Let $R_{t'} = \{b \in R | b \sim t'\}$. For any $b \in R_t$, $b \sim t$ and thus we have $b \sim t'$ and $R_t \subseteq R_{t'}$. Furthermore, $t'' = OR(R_{t'})$ would be in T after the Filtration phase of Algorithm 1 and t'' is compatible with every read in R_t.

During the Condensation phase of Algorithm 1, if t'' is not removed, the theorem holds. Otherwise, there must be another $t''' \in T$ such that all reads compatible with t'' are also compatible with t'''. In other words, all reads in R_t would be compatible with t'''. ■

2.2.3 The LASSO approach of estimating isoform expression levels

2.2.3.1 The mathematical model of RNA-Seq reads

Typical *alternative splicing (AS)* events include alternative $5'$ (or $3'$) splice sites, exon skipping, intron retention, mutually exclusive exons, *etc.*, but all these events can be dealt with in a unified mathematical model where a gene is partitioned into a sequence of *expressed segments* (or simply *segments*) based on exon-intron boundaries

[18]. More precisely, a gene is divided into a set of segments such that every segment is a continuous region in the reference genome uninterrupted by exon-intron boundaries. Then, a given set of candidate isoforms $T = \{t^1, t^2, \ldots, t^N\}$ for a gene can be represented as a binary matrix $A = (a_{ij})_{N \times M}$, where M is the number of segments of the gene. Each isoform corresponds to a row in this matrix such that $a_{ij} = 1$ if isoform t^i includes the jth segment, and 0 otherwise.

If we assume that a read is uniformly sampled from expressed isoforms, then the number of reads falling into each segment follows a binomial distribution, which can be approximated by a Poisson distribution [38] or Gaussian distribution [18] if the number of sequenced reads is large and the length of segments is small compared with the length of the reference genome. As a result, the expected number of reads falling into the jth segment, x_j, is proportional to both the segment length l_j and the sum of the expression levels of all isoforms containing the jth segment [38, 18]:

$$x_j = l_j \sum_{i=1}^{N} a_{ij} q_i \tag{2.1}$$

where q_i, the expected number of reads per base in isoform t^i, represents the expression level of t^i. Note that the expression level of an isoform can also be measured as RPKM (*i.e.*, Reads Per Kilobase of exon model per Million mapped reads, [58]). If there are totally E mapped reads, then an isoform t^i with expression level x_i has an expression level (in RPKM) $10^9 x_i / E$.

Notice that compared with the traditional multivariate regression model, the intercept is zero since we expect no read falling into the jth segment if none of the isoforms contain the segment, or if the expression levels of these isoforms are all zero.

We observe that the above model simplifies the real situation since the real

RNA-Seq data is biased. For example, because of the sequencing errors and repeat sequences in the reference genome, it is sometimes hard to decide whether a read really comes from a certain gene or exon (*i.e.*, the so called multi-read problem, which has been studied recently in [61]). Recent studies on RNA-Seq data also show that the above binomial model of read distribution may be an over-simplification [46, 69]. Some more complicated approaches have been proposed instead to handle different biases, such as using generalized Poisson distribution [79], considering the locality of bases [46], applying "effective length normalization" [69, 42], *etc.* In particular, the "effective length normalization" model can be easily incorporated in our model, by replacing the segment length l_j in Equation (2.1) with the "effective" segment length l'_j, where the length is calibrated by considering repeat sequences in the reference genome [42]. The issue of RNA-Seq biases will be discussed in detail in Chapter 3.

2.2.3.2 The LASSO approach

Given all mapped short reads and candidate isoforms of a gene, the expression levels of the candidate isoforms ($Q = \{q_1, \dots q_N\}$) can be estimated by minimizing the following residual sum of squares:

$$\begin{aligned} f(Q) &= \sum_{j=1}^{M} (\frac{x_j}{l_j} - \sum_{i=1}^{N} a_{ij} q_i)^2 \\ s.t.\ q_i &\geq 0, 1 \leq i \leq N \end{aligned} \tag{2.2}$$

However, such an approach may have several potential problems. For example, for a large value of N and a small value of M, the solution is not unique. It is also possible that a large number of estimated expression levels are small non-zero values which damage the interpretability. To address this latter problem, IsoInfer enumerates

combinations of isoforms and chooses a minimum set of isoforms such that the error $\sum_{j=1}^{M}(\frac{x_j}{l_j} - \sum_{i=1}^{N} a_{ij}q_i)^2$ is in a specified range. To deal with an exponential number of subsets of candidate isoforms, IsoInfer has to adopt several heuristics to make the algorithm practical. Also, some "shrinkage" methods which restrict the scale of Q can be used, like ridge regression [31], LASSO (or its variations like LARS [16], elastic-net [103], *etc*).

To achieve the minimization of interpretation without going through the exhaustive enumeration step in IsoInfer, we propose a new algorithm, called IsoLasso, based on LASSO. The LASSO approach minimizes the following objective function which seeks a balance between minimizing the overall error and minimizing the number of expressed isoforms:

$$f(Q) = \sum_{j=1}^{M}(\frac{x_j}{l_j} - \sum_{i=1}^{N} a_{ij}q_i)^2 + \lambda \sum_{i=1}^{N} |q_i| \tag{2.3}$$

The sparsity of variables, *i.e.*, minimizing the number of isoforms with non-zero expression levels, is obtained through the addition of an L1 normalization term, $\lambda \sum_{i=1}^{N} |q_i|$, to the original sum of squares. Since the expression level of each isoform should be non-negative, the above objective function leads to the following quadratic programming (QP) problem:

$$\begin{aligned} \min f(Q) &= \sum_{j=1}^{M}(\frac{x_j}{l_j} - \sum_{i=1}^{N} a_{ij}q_i)^2 + \lambda \sum_{i=1}^{N} q_i \\ s.t. \quad q_i &\geq 0,\ 1 \leq i \leq N \end{aligned} \tag{2.4}$$

which is equivalent to the following "constrained form" [85]:

$$
\begin{aligned}
\min f(Q) &= \sum_{j=1}^{M}(\frac{x_j}{l_j} - \sum_{i=1}^{N} a_{ij}q_i)^2 \\
s.t. \quad q_i &\geq 0,\ 1 \leq i \leq N \\
\sum_{j=1}^{N} q_i &\leq \gamma
\end{aligned}
\tag{2.5}
$$

The parameter λ (or γ) controls the number of isoforms with non-zero expression levels in the solution. In the constrained form of LASSO (Equation (2.5)), a larger value of γ will exert less restriction on the values of Q, which prefer a smaller sum of squares but more non-zero expression levels. In practice, a proper value of γ is selected via the "regularization path" [63], where several values of γ, $\gamma_1, \ldots \gamma_k$, are examined. If the values of the objective function in Equation (2.5) and the number of non-zero variables are $e_1, \ldots e_k$ and $L_1, \ldots L_k$, respectively, in these trials, then we define

$$
i^* = \underset{1 \leq i \leq k}{\operatorname{argmin}} \{L_i : e_i \leq \beta * \min\{e_1, \ldots e_k\}\} \tag{2.6}
$$

and select $\gamma = \gamma_{i^*}$, where β is a user-controlled parameter.

2.2.4 Completeness requirement

To ensure completeness, *i.e.*, each segments (or junction) with mapped reads covered by at least one isoform, the sum of expression levels of all isoforms that contain this segment (or junction) should be strictly positive. Formally, we add additional constraints to the above QP:

$$\begin{aligned}
\min f(Q) &= \sum_{j=1}^{M}(\frac{x_j}{l_j} - \sum_{i=1}^{N} a_{ij}q_i)^2 && (2.7)\\
s.t. \qquad q_i &\geq 0,\ 1 \leq i \leq N \\
\sum_{i=1}^{N} q_i &\leq \lambda \\
\sum_{i=1}^{N} q_i a_{ij} &\geq p, \text{ if segment } j \text{ has mapped reads} && (2.8)\\
\sum_{i=1}^{N} q_i a_{ij} a_{ik} \prod_{h=j+1}^{k-1} (1 - a_{ih}) &\geq p, \text{ if the junction between segments } j \text{ and } k \\
&\quad \text{contains mapped reads} && (2.9)
\end{aligned}$$

where p is a small positive threshold value to be decided empirically. The constraints (Equation (2.8) and Equation (2.9)) will ensure that all segments and junctions with mapped reads be covered by isoforms with positive expression levels in the solution of this QP.

The above QP problem can be solved by any standard QP solver, such as the "quadprog" function in Matlab [104]. In practice, however, if a gene contains too many segments and junctions, then there will be a large number of constraints involved, which make the above QP impractical to solve. As a compromise, we introduce the above constraints only for segments (or junctions) with expression levels above a certain threshold.

2.3 Experimental results

2.3.1 Simulated mouse RNA-Seq data

We use UCSC mm9 gene annotation [36] to generate simulated single-end and paired-end reads. An *in silico* RNA-Seq data generator, Flux Simulator [74], is used to generate simulated reads. Flux Simulator first randomly assigns an expression level to every isoform in the annotation, and then simulates the library preparation process in a typical RNA-Seq experiment (including reverse transcription, fragmentation, size selection, *etc*). After that, reads are generated in the sequencing step. Various error models can be incorporated in these steps; but in our simulations, only error-free reads are simulated to compare the performance of different algorithms in the ideal situation.

The distribution of the expression levels of all 49409 isoforms in the UCSC mm9 gene annotation is plotted in Figure 2.3 (A).

2.3.1.1 Matching criteria

All assembled isoforms (referred to as "candidate isoforms") are matched against all known isoforms in the annotation (referred to as "benchmark isoforms"). Two isoforms match iff:

1. They include the same set of exons; and

2. All internal boundary coordinates (*i.e.*, all the exon coordinates except the beginning of the first exon and the end of the last exon) are identical.

Two single-exon isoforms match iff the overlapping area occupies at least 50% the length of each isoform.

Following [18], we use *sensitivity*, *precision* and *effective sensitivity* to evaluate the performance of different programs. Sensitivity and precision are defined as follows:

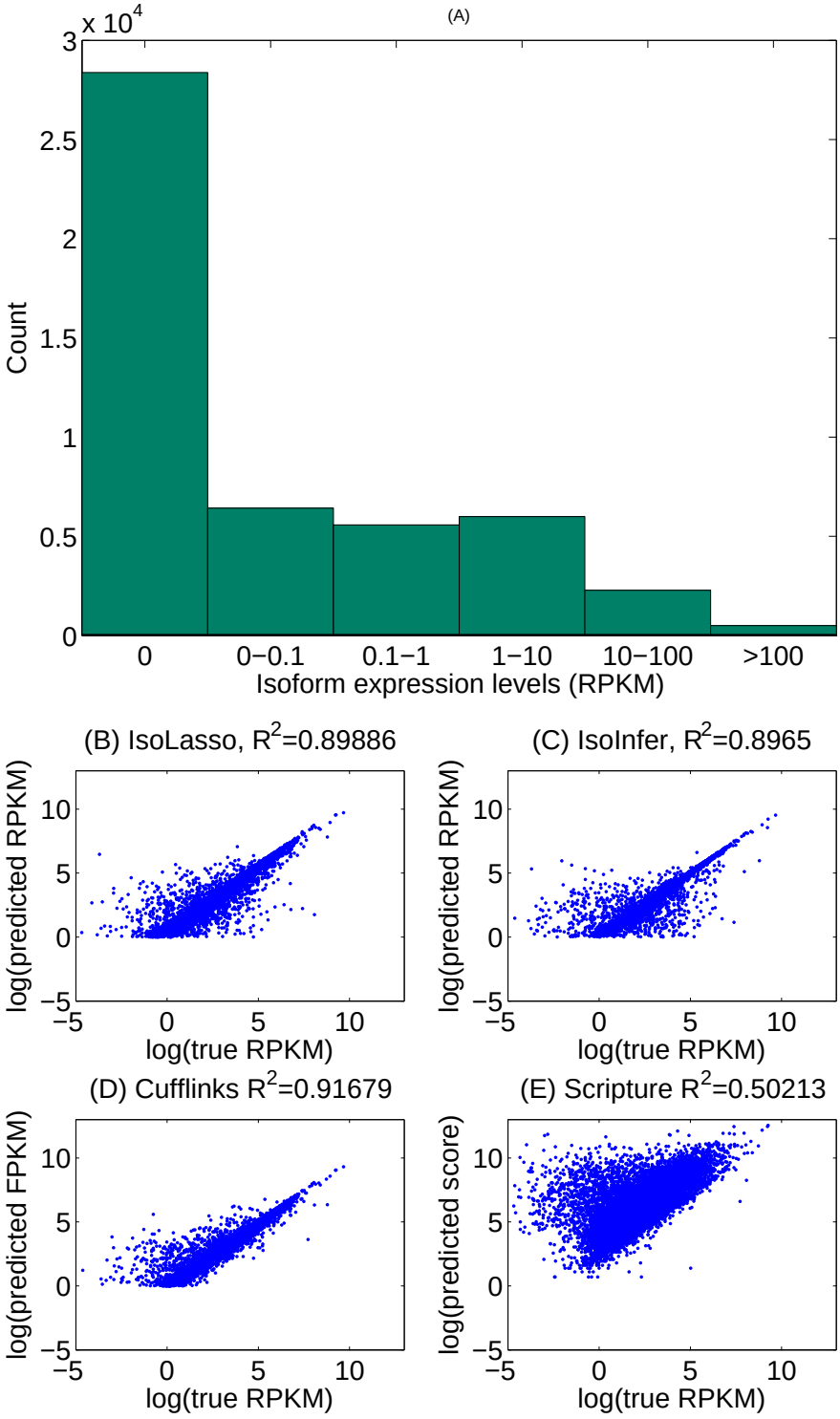


Figure 2.3: The distribution of simulated isoform expression levels (A), and the expression level estimation accuracies of IsoLasso (B), IsoInfer without TSS/PAS (C), Cufflinks (D), and Scripture (E). Note that Scripture computes a "weighted score" instead of RPKM value for each predicted isoform.

if K out of M benchmark isoforms match K' out of N candidate isoforms, then

$$\text{sensitivity} = K/M \tag{2.10}$$

$$\text{precision} = K'/N \tag{2.11}$$

Note that several candidate isoforms may match the same benchmark isoform.

Effective sensitivity is calculated based on the isoforms satisfying *Condition I* defined in [18]. Isoforms satisfying Condition I are those with all segment junctions covered by at least one short read. If there are S benchmark isoforms satisfying Condition I and K of them are matched, then

$$\text{effective sensitivity} = K/S \tag{2.12}$$

Intuitively, isoforms satisfying Condition I are those that are relatively easy to predict, since all their segment junctions are covered by short reads. It is shown in [18] that an isoform with a higher expression level is more likely to satisfy this condition.

2.3.1.2 Sensitivity, precision and effective sensitivity

In this section, we use the sensitivity, precision and effective sensitivity defined above to compare IsoLasso with IsoInfer (version V0.9.1, downloaded from website `http://www.cs.ucr.edu/~jianxing/IsoInfer.html`), Cufflinks (version 0.9.1, downloaded from website `http://cufflinks.cbcb.umd.edu`), and Scripture (beta version, downloaded from website `http://www.broadinstitute.org/software/scripture/home`). We use TopHat [86] to map all simulated short reads with multi-reads discarded. Then, the read mapping information serves as the input for all four programs. Since IsoInfer is

based on the assumption that the boundaries of all genes and exons are known, we infer exon boundaries from mapped junction reads using TopHat and infer gene boundaries by clustering overlapping mapped reads. Note that IsoInfer is actually designed to take advantage of any known transcription start site and poly-A site (TSS/PAS) information, although it also works without such information. Since the other three programs do not use the TSS/PAS information, neither does IsoInfer use such information in the comparison.

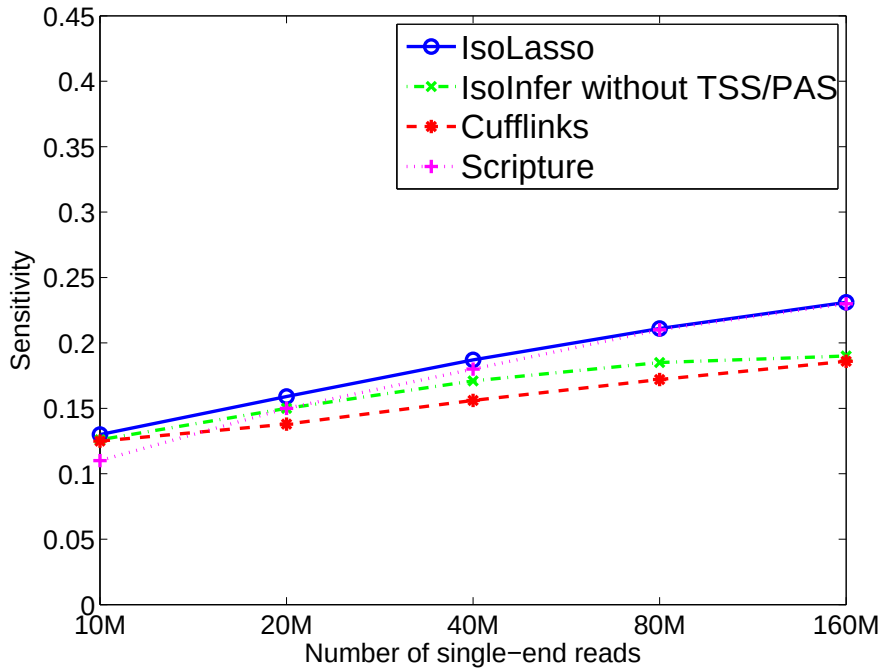


Figure 2.4: The sensitivity of all programs on simulated single-end reads.

Figure 2.4-2.6 and Figure 2.7-2.9 plot the sensitivity, precision and effective sensitivity using various numbers of single-end and paired-end reads, respectively. On single-end reads, all transcriptome assembly tools achieve a higher sensitivity and precision as more reads are used for the assembly. Among them, IsoLasso outperforms

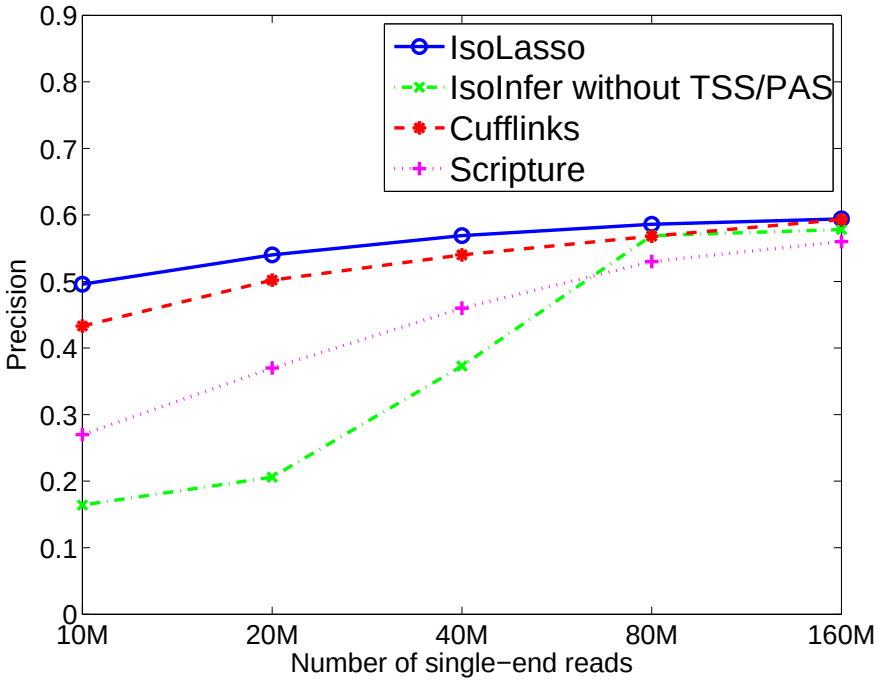


Figure 2.5: The precision of all programs on simulated single-end reads.

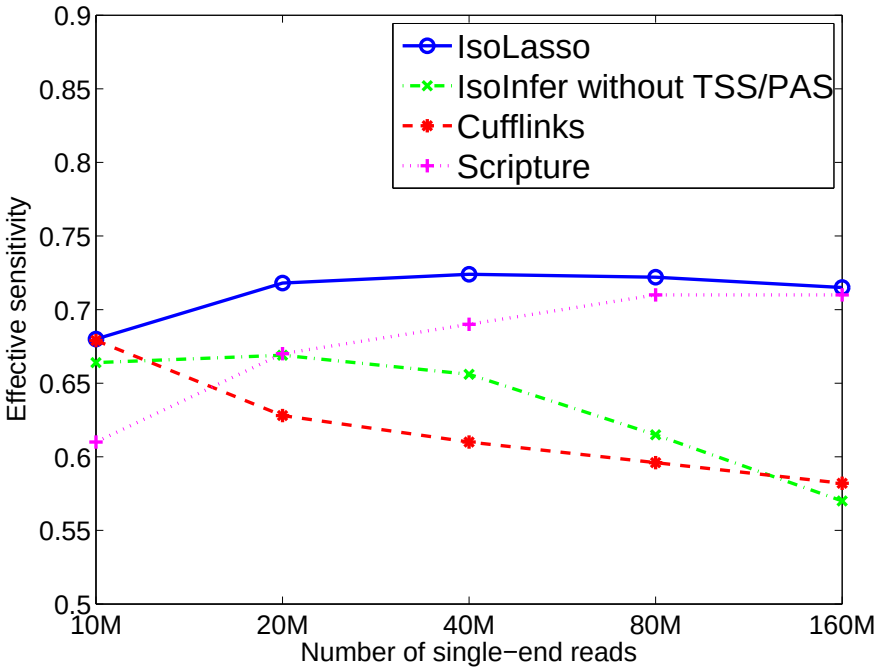


Figure 2.6: The effective sensitivity of all programs on simulated single-end reads.

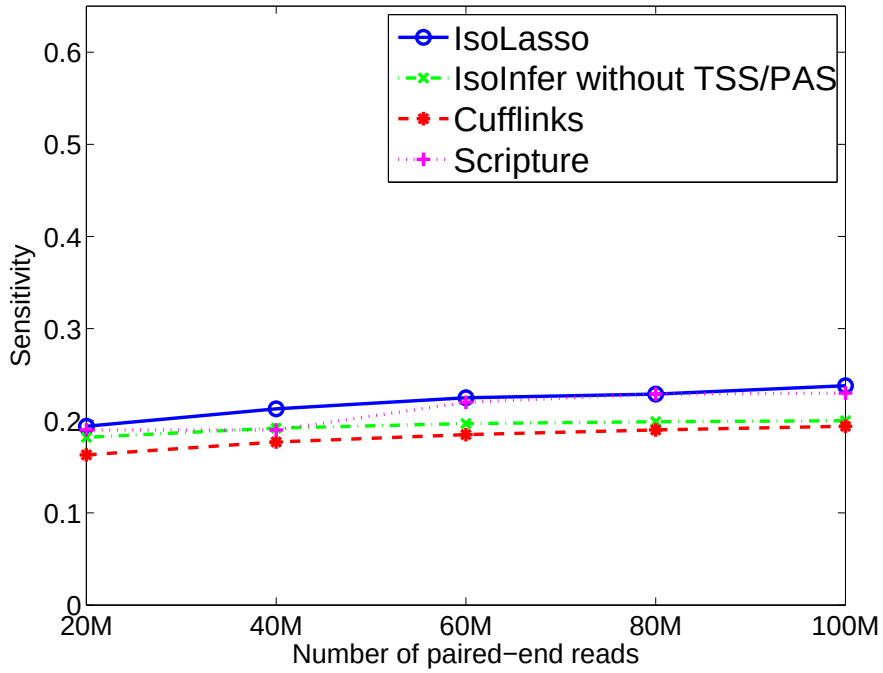


Figure 2.7: The sensitivity of all programs on simulated paired-end reads.

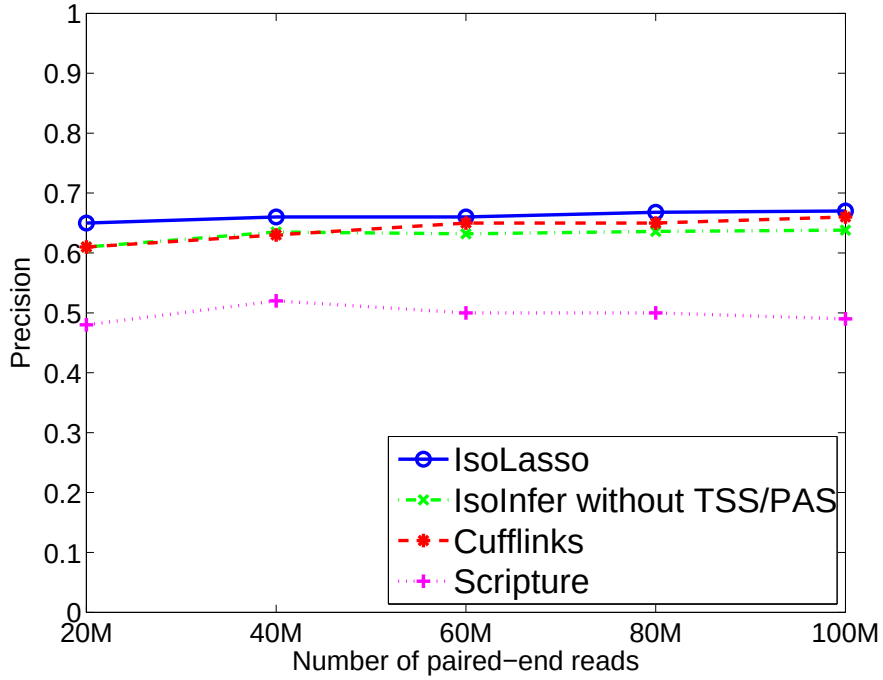


Figure 2.8: The precision of all programs on simulated paired-end reads.

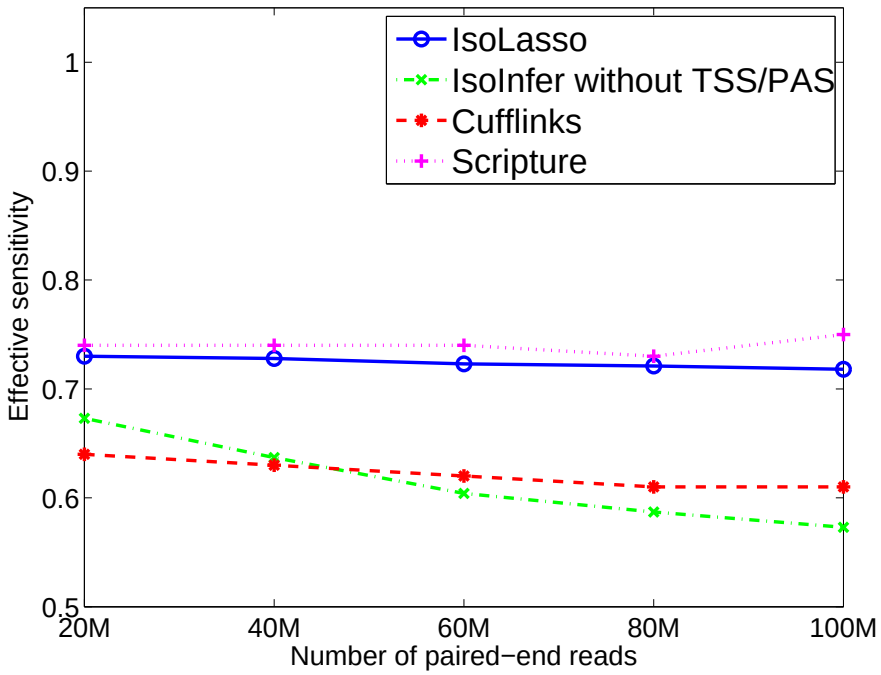


Figure 2.9: The effective sensitivity of all programs on simulated paired-end reads.

all other programs with respect to all three criteria. This is perhaps because IsoLasso is able to maintain a good interpretation by filtering out many lowly expressed false predictions (which leads to a high precision), while keeping highly expressed isoforms and a high effective sensitivity. Scripture seems to benefit the most when more reads are available. Also, IsoInfer exhibits a sharp increase in precision from less than 20% to more than 50%, at the cost of decreased effective sensitivity (by about 10%).

On paired-end reads, IsoLasso also achieves the best precision and sensitivity as well as a good balance between precision and effective sensitivity. However, it is surprising to see that when the number of paired-end reads increases from 20M to 100M, a less than 10% increase in sensitivity and precision is observed for all the algorithms. Also, none of the algorithms have a significant increase in effective sensitivity. In fact, both Cufflinks and IsoInfer see their effective sensitivities decreased a bit when more single-end and paired-end reads are used. This is because more benchmark isoforms

would satisfy Condition I of [18] as the sequencing depth increases. In this case, more isoforms are expected to be expressed for each gene, which result in a more complicated overlap graph for Cufflinks and a larger search space for IsoInfer.

Cufflinks reaches a high precision by filtering out many lowly expressed isoforms, but this sacrifices the effective sensitivity. On the other hand, Scripture achieves the highest effective sensitivity by enumerating all possible paths in the connectivity graph, but its precision is low since many of the paths are false positives.

2.3.1.3 Expression level estimation

All programs estimate the expression levels of predicted isoforms using different measures. Both IsoLasso and IsoInfer estimate expression levels in RPKM [58], while Cufflinks uses the term FPKM (expected number of Fragments Per Kilobase of transcript sequence per Millions base pairs sequenced) [87]. Scripture does not predict expression levels directly; instead, it computes a "weighted score" for each isoform to indicate how likely the isoform is expressed.

Fig. 2.3 (B) $\sim$ (E) plot the predicted and true expression levels for all predicted isoforms which are matched to the benchmark isoforms and have expression levels > 1 RPKM, using the 80M paired-end read dataset. The plots show that IsoLasso, IsoInfer and Cufflinks estimate expression levels quite accurately (the squared correlation coefficient between the predicted and true expression levels is $R^2 > 0.89$), while the "weighted score" of Scripture does not directly reflect the true expression level of isoforms ($R^2 = 0.50$). Cufflinks shows the highest prediction accuracy in expression level estimation ($R^2 = 0.91$) partly because it uses an accurate iterative statistical model to estimate the expression levels [87], which could potentially be incorporated into our method as a refinement step.

2.3.1.4 More isoforms, more difficult to predict

Intuitively, genes with more isoforms are more difficult to predict. We group all the genes by their numbers of isoforms, and calculate the sensitivity and effective sensitivity of the algorithms on genes with a certain number of isoforms as shown in Figure 2.11-2.12. Figure 2.10 shows the total number of isoforms and isoforms satisfying Condition I ([18]) grouped by the number of isoforms per gene.

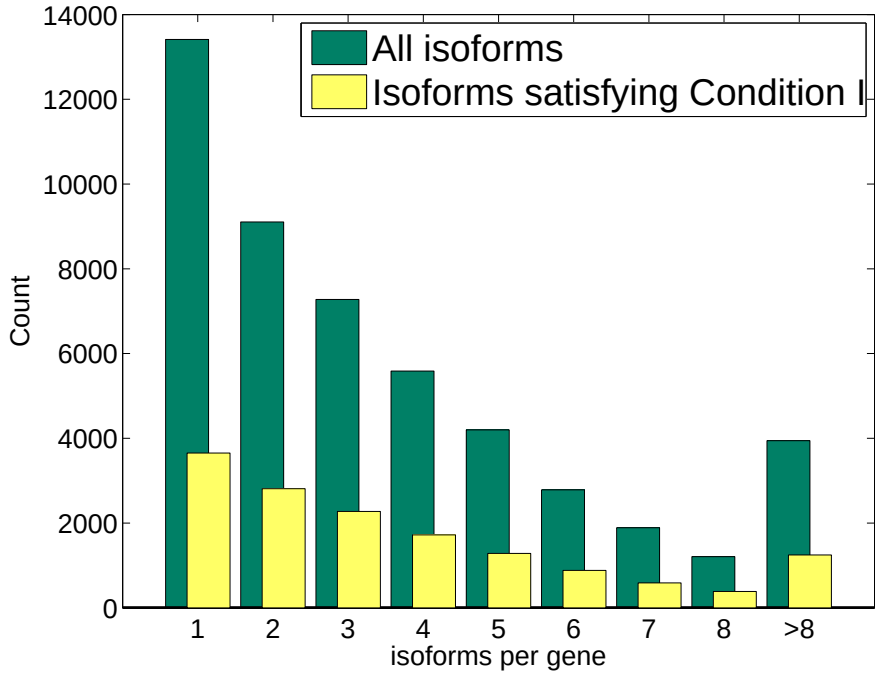


Figure 2.10: The total number of isoforms and isoforms satisfying Condition I.

Figure 2.10-2.12 shows that genes with more isoforms are more difficult to predict correctly, as both sensitivity and effective sensitivity decrease for genes with more isoforms. IsoLasso and Scripture outperform IsoInfer and Cufflinks in general. IsoLasso has a higher sensitivity and effective sensitivity on genes with at most 5 isoforms, but Scripture catches up with IsoLasso on genes containing more than 5 isoforms.

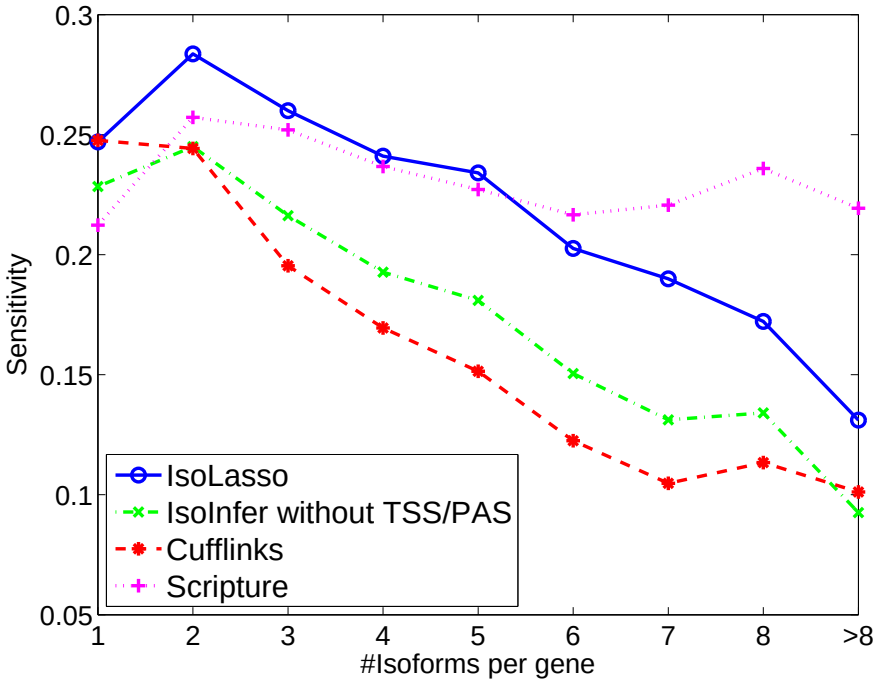


Figure 2.11: The sensitivity of the algorithms grouped by the number of isoforms per gene. Here, 100M paired-end reads are simulated.

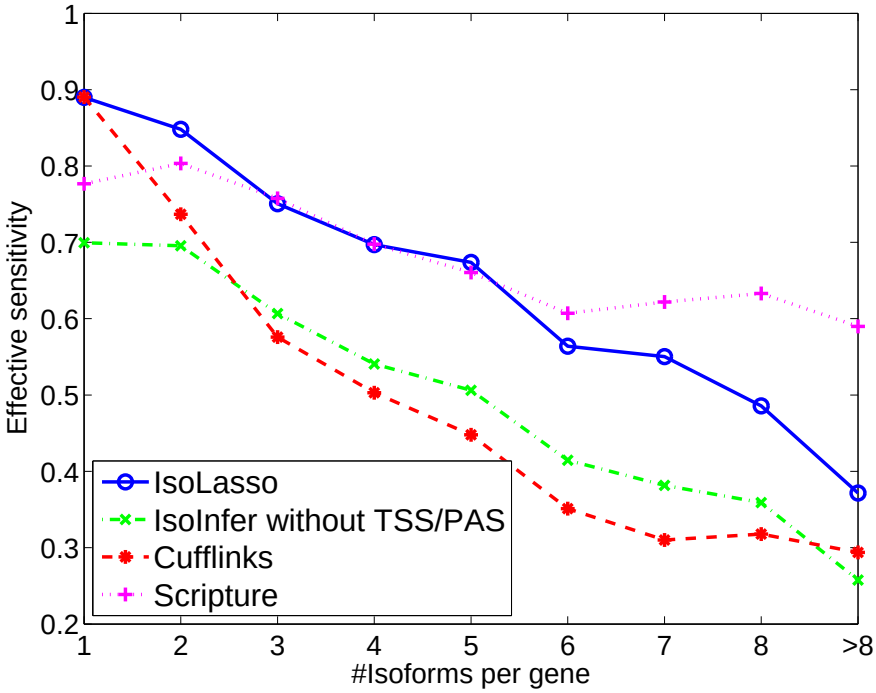


Figure 2.12: The effective sensitivity of the algorithms grouped by the number of isoforms per gene. Here, 100M paired-end reads are simulated.

2.3.1.5 Running time

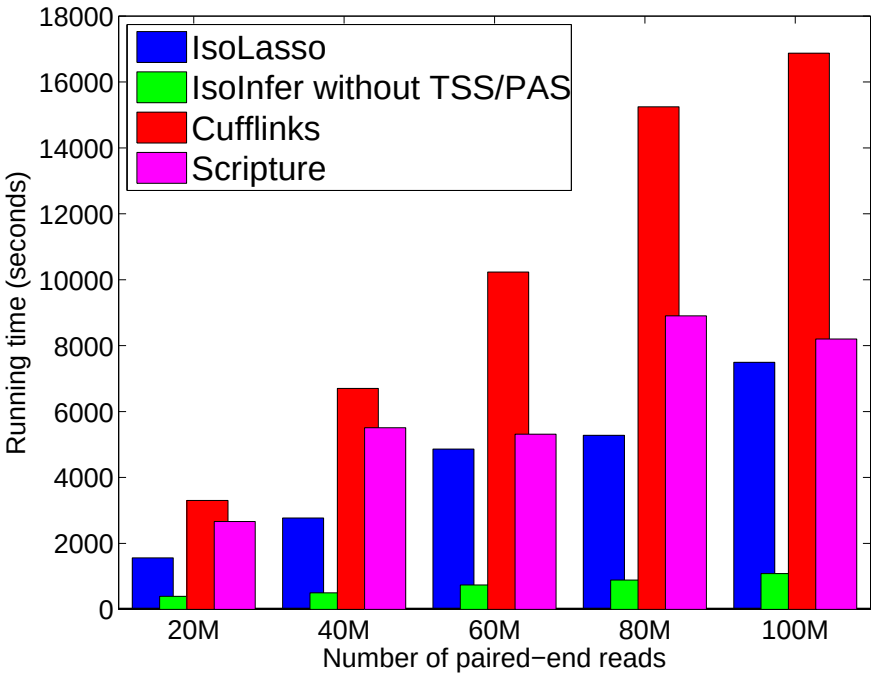


Figure 2.13: The running time for all the algorithms.

Figure 2.13 plots the running time of all four transcript assembly programs using various numbers of paired-end reads. The time for data preparation is excluded, including mapping reads to the reference genome and preparing required input files for both IsoLasso and IsoInfer. Surprisingly, although employing a search algorithm, IsoInfer runs much faster than that of any other algorithm. This is partly due to the heuristic restrictions that IsoInfer adopts to reduce the search space (*e.g.*, requiring the candidate isoforms to satisfy Condition I and some other conditions), and the programming languages used in each tool (IsoInfer, IsoLasso, Scripture and Cufflinks use C++, Matlab, Java, and Boost C++, respectively). All programs are run on a single 2.6 GHz CPU, but Cufflinks allows the user to run on multiple threads, which may substantially speed up the assembly process.

2.3.1.6 Comparision between the newest versions of IsoLasso and Cufflinks

Both IsoLasso and Cufflinks have been updating their source codes frequently since their first release in 2010. The performance difference of different versions of the same software may be huge due to various reasons, for example, fixed bugs and improved implementations of the algorithm or the pre-processing/post-processing procedures. Here, we compare the performance of the latest versions of IsoLasso (version 2.5.0) and Cufflinks (version 1.3.1) on simulated datasets. IsoLasso was originally implemented in Matlab but was rewritten in C++ later, so the performance of different IsoLasso implementations is also compared.

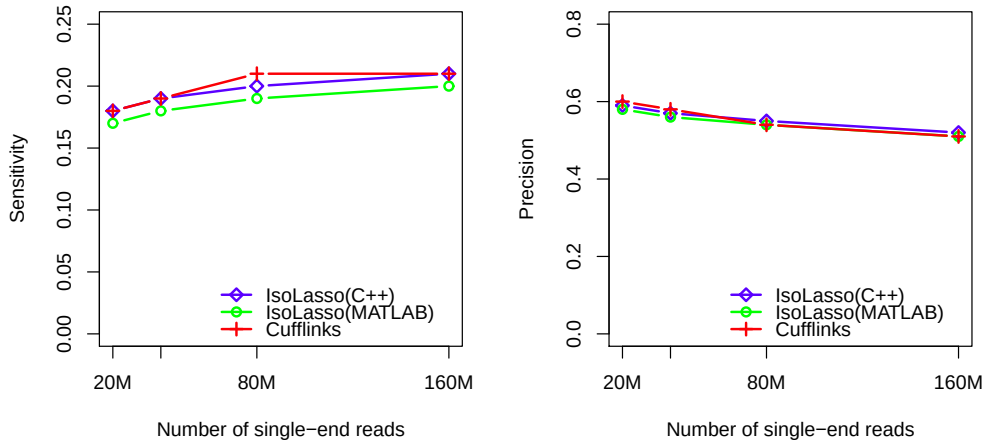


Figure 2.14: The sensitivity (left) and precision (right) of the newest versions of IsoLasso and Cufflinks on simulated single-end reads.

Figure 2.14 and Figure 2.15 plot the sensitivity and precision of IsoLasso and Cufflinks using various numbers of simulated single-end and paired-end reads, respectively. For simulated single-end data, Cufflinks achieves similar sensitivity and precision as IsoLasso, but on paired-end reads, Cufflinks outperforms IsoLasso by a 2%-4% higher

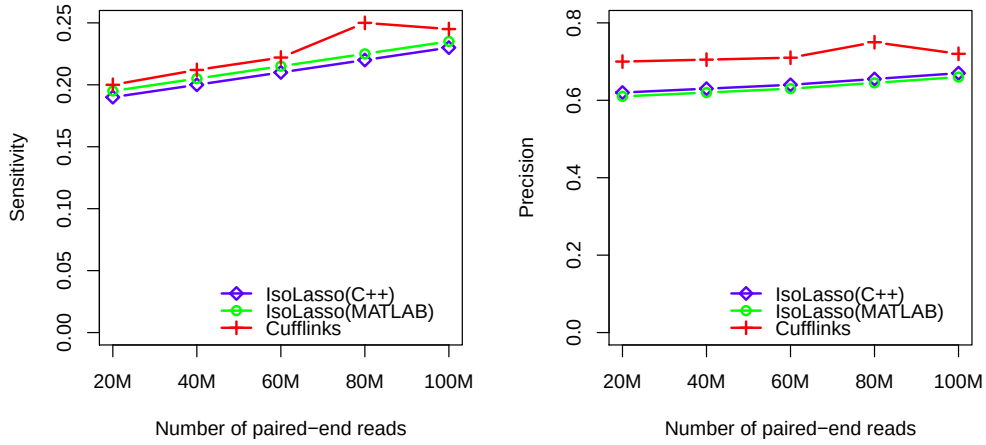


Figure 2.15: The sensitivity (left) and precision (right) of the newest versions of IsoLasso and Cufflinks on simulated paired-end reads.

sensitivity and a 3%-7% higher precision. Since the core algorithm of Cufflinks does not change much between different versions, the greatly improved precision may be partially credited to removing some lowly expressed isoforms from the prediction. Indeed, by adopting a similar approach, the sensitivity and precision of IsoLasso are close to Cufflinks (see Figure 2.16). The C++ version of IsoLasso has a higher sensitivity than the Matlab version, which may be due to a better implementation of the algorithm in the C++ version.

Figure 2.17 shows the prediction accuracy in terms of R^2 values for Cufflinks and IsoLasso (the C++ version). The R^2 values of both Matlab and C++ versions of IsoLasso are close, but the prediction accuracy of Cufflinks is lower compared to its earlier versions (see Figure 2.3). The reason is that Cufflinks adopted a new expression level calculation model that greatly overestimates the expression levels of some transcripts (the circles above the main diagonal in Figure 2.17 left). The length of these isoforms are short compared to the read length, and Cufflinks uses a fragment length model that

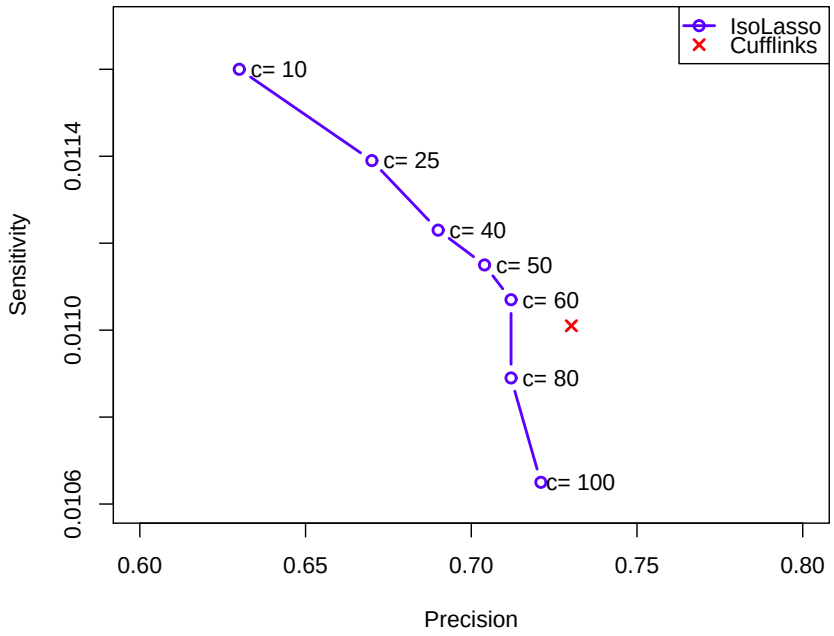


Figure 2.16: By filtering lowly expressed isoforms, IsoLasso reaches similar sensitivity and precision compared to Cufflinks. Here, 40 million paired-end reads are generated, and the sensitivity and precision are calculated based on UCSC known transcripts in human chromosome 1. c is a parameter in IsoLasso: the predicted isoforms will be filtered if the number of reads in the corresponding genes is smaller than c.

assumes short fragments (short DNA sequences after fragmentation and before sequencing) are rare. A similar observation of this behavior of Cufflinks is also reported recently in [43].

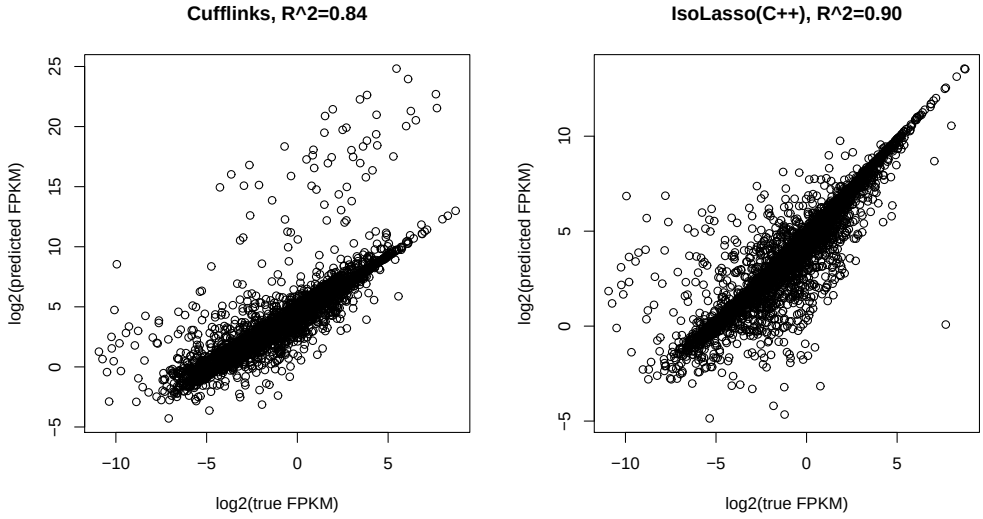


Figure 2.17: The expression level estimation accuracy, in terms of R^2 values, of Cufflinks (left) and IsoLasso (C++ version, right). The expression level estimations of IsoLasso Matlab version are not plotted since the R^2 value is close to the IsoLasso C++ version (0.89).

2.3.2 Real RNA-Seq data

Reads from two real RNA-Seq experiments are used to evaluate the performance of IsoLasso, Cufflinks and Scripture. We exclude IsoInfer from the comparison because its algorithm is similar to (and improved by, as seen from the simulation results) the algorithm of IsoLasso. One RNA-Seq read dataset is generated from the C2C12 mouse myoblast cell line ([87], NCBI SRA accession number SRR037947), and the other from human embryonic stem cells (Caltech RNA-Seq track from the ENCODE project [82], NCBI SRA accession number SRR065504). Both RNA-Seq datasets include 70 million and 50 million 75 bp paired-end reads which are mapped to the UCSC *mus*

musculus (mm9) and *homo sapiens* (hg19) reference genomes using Tophat [86], respectively.

Isoforms inferred by programs IsoLasso, Cufflinks and Scripture are first matched against the known isoforms from mm9 and hg19 reference genomes. There are a total of 11484 and 12193 known mouse and human isoforms recovered by at least one program, respectively (Figure 2.18 (A) and (B)). Among these isoforms, 4485 (39%) and 4274 (35%) isoforms are detected by all programs, while 8204 (71%) and 8084 (66%) isoforms are detected by at least two programs. These numbers show that, although there is a large overlap (more than 60%) among the known isoforms recovered by these programs, each program also identifies a substantially large number of "unique" isoforms. Such "uniqueness" of each program is shown more clearly if we compute the overlap between their predicted isoforms directly (see Figure 2.18 (C) and (D)). Each of the three programs predicts more than 40,000 isoforms on both dataset, but only shares 2% to 20% isoforms with other programs. About 49.5% of the mouse isoforms (46% in human) inferred by IsoLasso are also predicted by at least one of other two programs, which is substantially higher than Cufflinks (27.7% in mouse and 38.4% in human) and Scripture (4.6% in mouse and 7.4% in human). This may indicate that IsoLasso's prediction is more reliable than those of Cufflinks and Scripture since it receives more support from other (independent) programs.

Note that among all the isoforms inferred by IsoLasso, Cufflinks and Scripture, 9741 mouse isoforms and 11381 human isoforms are predicted by all three programs. These isoforms could be considered as "high-quality" ones. However, fewer than a half of these "high-quality" isoforms (4485 in mouse and 4274 in human) could be matched to the known mouse and human isoforms (see Figure 2.18 (A) and (B)). This suggests that the current genome annotations of both mouse and human are still incomplete. An

example of the "high-quality" isoforms is shown in Figure 2.18 (E). Here, an isoform with an alternative 5′ end of gene Tmem70 in mouse is predicted by all three programs but cannot be found in the mm9 RefSeq annotation or GenBank mRNAs (track not shown in the figure).

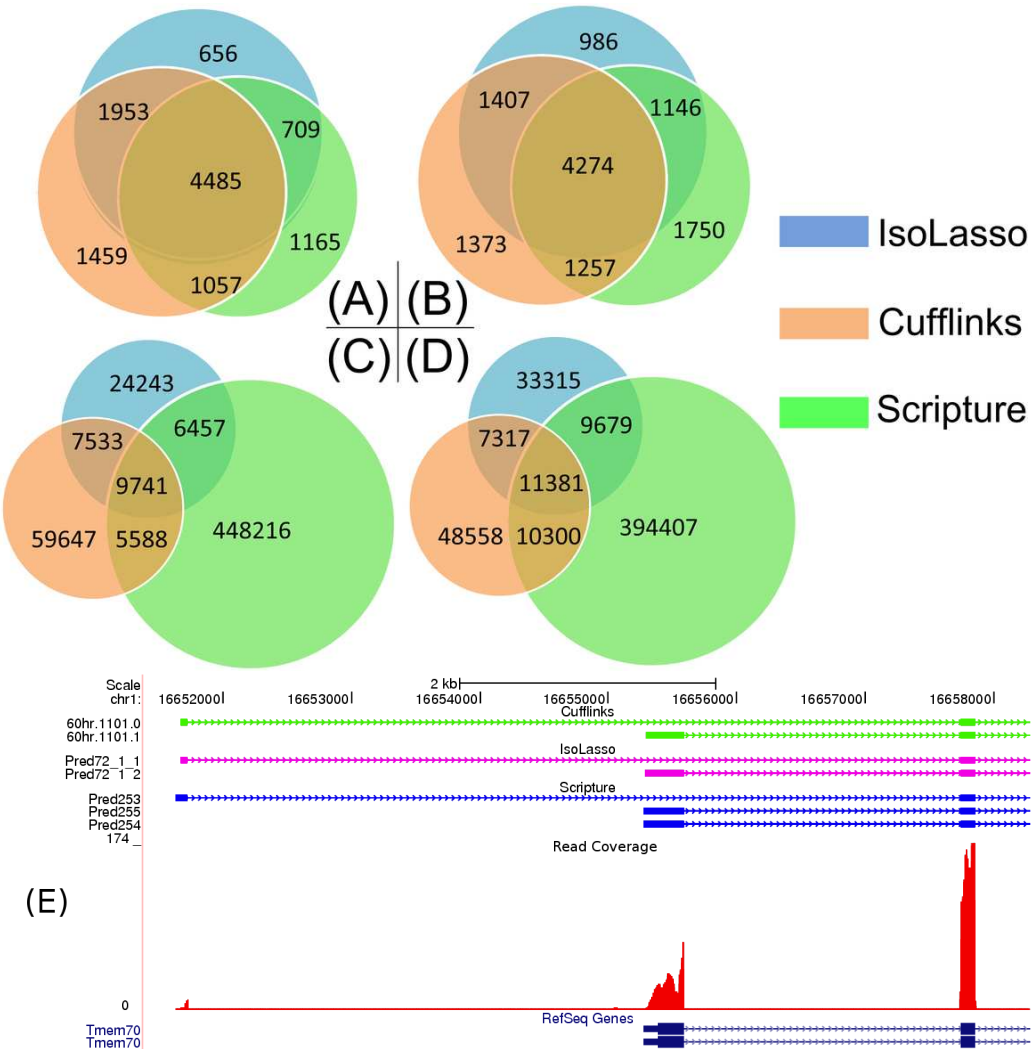


Figure 2.18: The numbers of matched known isoforms of mouse (A) and human (B), and the numbers of predicted isoforms of mouse (C) and human (D), assembled by IsoLasso, Cufflinks and Scripture. (E) shows an alternative 5' start isoform of gene Tmem70 in mouse C2C12 myoblast RNA-Seq data [87]. This isoform does not appear among the known isoforms, but is detected by IsoLasso, Cufflinks and Scripture. Tracks from top to bottom: Cufflinks predictions, IsoLasso predictions, Scripture predictions, the read coverage, and the Tmem70 gene in the mm9 RefSeq annotation.

2.4 Conclusion

RNA-Seq transcriptome assembly is a challenging computational biology problem that arises from the development of second generation sequencing. In this paper, we proposed three fundamental objectives/principles in the transcriptome assembly: prediction accuracy, sparsity, and completeness. We also presented IsoLasso, an algorithm based on the LASSO approach that seeks a balance between these objectives. Experiments on simulated and real RNA-Seq datasets show that, compared with the existing transcript assembly tools (IsoInfer, Cufflinks and Scripture), IsoLasso is efficient and achieves the best overall performance in terms of sensitivity, precision and effective sensitivity compared to IsoInfer and Scripture. The latest version of IsoLasso achieves comparable performance to the latest version of Cufflinks.

Chapter 3

Transcriptome Assembly and Isoform Expression Level Estimation from Biased RNA-Seq Reads

3.1 Introduction

In this chapter, we propose a statistical framework based on the quasi-multinomial distribution model [9, 11] to capture RNA-Seq biases, including positional, sequencing and mappability biases. The framework allows us to develop an expectation-maximization (EM) algorithm [13] for both transcriptome assembly and isoform abundance level estimation from biased RNA-Seq data. Compared with other algorithms in the literature that use sophisticated probabilistic generative models to handle biases [46, 70], our EM algorithm uses a single parameter to capture the property of RNA-Seq biases of different

types. Utilizing the isoform enumeration algorithm of IsoLasso [47], the EM algorithm assembles isoforms and estimates their abundance levels at the same time. Moreover, both principles of *prediction accuracy* and *interpretation* (or "sparsity") considered in [47] are achieved in the assembly.

The rest of this chapter is organized as follows. The statistical framework and the EM algorithm are introduced in Section 3.2 and 3.3.2, and in Section 3.4, we demonstrate the superior performance of our EM algorithm compared with other algorithms in the literature through simulated and real RNA-Seq experiments, and analyze the effects of RNA-Seq biases on both transcriptome assembly and isoform abundance level estimation. This chapter is concluded in Section 3.5, and some additional remarks to the quasi-multinomial model are in Section 3.6.

3.2 The quasi-multinomial model for isoform abundance level estimation

3.2.1 The Poisson model and the generalized Poisson (GP) model

We use the mathematical model of isoforms in Chapter 2. Basically, for a gene G with M segments, if G induces N isoforms (denoted as $T = \{t^1, \ldots, t^N\}$), then these isoforms can be represented as an $N \times M$ binary matrix $A = \{a_{ij}\}$, where $a_{ij} = 1$ if isoform t^i includes exon (or expressed segment) j, and 0 otherwise.

Let X_j be the random variable of the read counts falling into exon j. Under the assumption that a read r is sampled uniformly from an isoform (the "Poisson assumption"), X_j follows a Poisson distribution with parameter λ_j proportional to the length of exon j and the total abundance level of all isoforms containing exon j [38]:

$$P(X_j = x_j) = \frac{e^{-\lambda_j}\lambda_j^{x_j}}{x_j!}. \tag{3.1}$$

$$\lambda_j = Wl_j \sum_{i=1}^{N} a_{ij} q_i = \sum_{i=1}^{N} c_{ij} q_i \tag{3.2}$$

where W denotes the total number of sequenced reads, c_{ij} a known constant, and q_i the abundance level of isoform t^i. The abundance level is usually measured by RPKM [58] or FPKM [87], and can be estimated by maximizing the joint probability of observing $x_1, \ldots, x_M$ reads in M exons, as proposed in [38].

To model RNA-Seq biases, [79] further uses a *Generalized Poisson (GP)* distribution $GP(\lambda_j, \rho)$ [9] to model X_j:

$$P(X_j = x_j) = \lambda_j(\lambda_j + x_j\rho)^{x_j - 1} e^{-x_j\rho - \lambda_j}/x_j! \tag{3.3}$$

where $-1 \leq \rho \leq 1$ is the bias parameter to account for the biases in each read count x_j. The Poisson distribution is a special case of GP distribution where $\rho = 0$. The GP model can also be used to estimate isoform expression levels similar to the approach in [38], since the sum of M GP distributions, $GP(\lambda_1, \rho)$,..., $GP(\lambda_M, \rho)$, also follows a GP distribution $GP(\sum_{j=1}^{M} \lambda_j, \rho)$.

However, this "count-only" approach (and also the approach in [38]) uses only the information of read counts (*i.e.*, $x_1, \ldots, x_M$ in Equation (3.3)) and it does not consider the fact that a read may come from different isoforms with different probabilities due to the sampling biases. In the following, we develop a quasi-multinomial model [11], which generalizes the GP model, to capture biases in RNA-Seq data.

3.2.2 The quasi-multinomial model

Consider a single-end (or paired-end) read r_j of length L that is mapped to exon j of length l_j from gene G. Denote $\theta_i = P(t^i)$ as the prior probability that read r_j comes from t^i with the constraint $\sum_{i=1}^{N} \theta_i = 1$. Clearly, θ_i depends on q_i, the abundance level of t^i, and the abundance level q_i in Equation (3.2) can be replaced by θ_i as:

$$q_i = \frac{R\theta_i}{Wl_i} \tag{3.4}$$

We may think of the process of sampling r_j as follows: one of the isoforms t^i is first randomly selected with probability θ_i and then a read r_j belonging to exon j is sampled from t^i with probability $P(r_j|t^i)$. To model positional (and other) biases, the probability $P(r_j|t^i)$ can be defined as a distribution $f(k_{i,j})$ depending on the location $k_{i,j}$ of r_j in t^i. Note that if f is the uniform distribution, then

$$P(r_j|t^i) = \frac{a_{ij}(l_j - L + 1)}{L_i - L + 1} \tag{3.5}$$

where L_i is the length of t^i. $f(k_{i,j})$ can also be an exponential function to model the RNA degradation process which plays an important role in the formation of the positional bias [89]. For paired-end reads, $f(k_{i,j})$ can be modified to incorporate the probability distribution of the span of the read pairs (see Section 3.6.3).

Several strategies can be used to construct a non-uniform distribution f. For example, a non-uniform positional distribution can be determined empirically and incorporated into f [99]. Also, the "effective length" of isoforms excluding repeat regions of the reference genome can be used in Equation (3.5) to handle mappability biases [69]. [1]

[1] This latter technique is not yet realized in our implementation.

In our implementation, we use the method in [99] to model the non-uniform distribution of f.

The probability of observing read r_j is thus

$$P(r_j) = \sum_{i=1}^{N} P(r_j|t^i)P(t^i) = \sum_{i=1}^{N} \theta_i f(k_{i,j}) \tag{3.6}$$

and the joint probability of observing the R reads mapped to gene G follows a quasi-multinomial distribution:

$$P(R|\theta,\tau) = \binom{R}{x_1,\ldots,x_M}(1+R\tau)^{1-R}\prod_{j=1}^{M} P(r_j)(P(r_j)+\tau x_j)^{x_j-1} \tag{3.7}$$

where $\tau > -1/R$ is the bias parameter (similar to the parameter ρ in the GP distribution). The quasi-multinomial distribution reduces to multinomial distribution when $\tau = 0$. The value of τ indicates how read counts differ from a multinomial distribution: if $\tau > 0$ then too many reads are observed (called "over-dispersion") and if $\tau < 0$ (called "under-dispersion"), fewer reads are observed. Similar to the relationship between multinomial and Poisson distributions, Equation (3.7) can be approximated by a product of M GP distributions [11], and thus finding an optimal τ is equivalent to finding an optimal ρ in the GP model (See Section 3.6.2).

3.3 Component elimination EM

3.3.1 Transcriptome assembly

We use the candidate isoform enumeration algorithm introduced in IsoLasso (Algorithm 1), which is proven to generate the same set of candidate isoforms considered

by Cufflinks [87]. The algorithm first enumerates all possible paths in the *connectivity graph* [26] constructed from the mapped reads. Then two additional steps are applied to remove infeasible paths and non-maximal paths.

IsoLasso uses the LASSO algorithm [85] to select candidate isoforms and estimate their abundance levels. However, the LASSO algorithm is solved by constrained quadratic programming which could be very slow if many constraints are imposed. Moreover, it is unable to handle biases in RNA-Seq data. We will develop an expectation-maximization (EM) algorithm (called *component elimination EM*) in the next section based on the above quasi-multinomial model to select candidate isoforms and estimate their abundance levels from biased RNA-Seq data. Note that EM algorithms are routinely used in RNA-Seq data analysis, and several EM algorithms have been proposed in the literature to use information beyond read counts to improve the accuracy of isoform abundance level estimation. For example, multi-reads (*i.e.*, reads mapped to several locations of the reference genome) are utilized to estimate the abundance levels of isoforms [44] or homologous genes [61]. In [87, 70, 73], the distribution of distances between read pairs in paired-end RNA-Seq data is incorporated into the EM algorithms. Such information can be readily incorporated to our quasi-multinomial model (and thus our EM algorithm) (see Section 3.6.3 for more details).

3.3.2 Using negative Dirichlet distribution to achieve sparsity

It is commonly believed that a gene usually has only a few highly expressed isoforms [47]. For this reason, ensuring a good interpretation (or "sparsity") is critical in transcriptome assembly, as is discussed in Chapter 2. Generally speaking, in the context of EM algorithm, a good interpretation is to keep the number of *components* [19] (*i.e.*, the number of models whose probabilities are to be determined in the algorithm) as

small as possible. However, if the number of isoforms (or components) is large, the standard EM algorithm may deliver results that lack sparsity, *i.e.*, solutions with many components having small non-negative probabilities instead of solutions with only a few components having large probabilities while the others having zero probability [19]. To achieve sparsity, a negative Dirichlet prior distribution of θ is added multiplicatively to the quasi-multinomial likelihood function in Equation (3.7) [19, 4]:

$$P(\theta) \propto \prod_{i=1}^{N} \theta_i^{-\alpha} \tag{3.8}$$

where α is the negative Dirichlet parameter specified by the user. The negative Dirichlet distribution assigns a higher probability if one or more of the values of θ_i are closer to 0 (see Figure 3.1). Hence, solutions with fewer non-zero values of θ_i are preferred (see Figure 3.1).

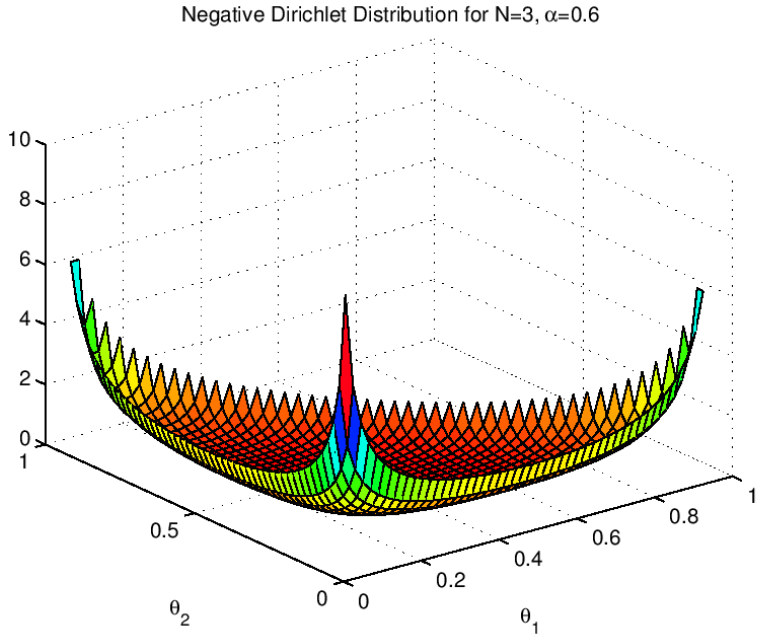


Figure 3.1: An example of negative Dirichlet distribution in Equation (3.8) with $N = 3$ and $\alpha = 0.6$. The values of θ_i satisfy the constraint $\theta_1 + \theta_2 + \theta_3 = 1$, and the density function increases when one of the variables is close to 1 and the others to 0.

3.3.3 The EM model

Combining both quasi-multinomial distribution (Equation (3.7)) and the negative Dirichlet distribution (Equation (3.8)), the joint log-likelihood function can be written as follows:

$$\begin{aligned} logP(R, Z, \theta, \tau) &= logP(R|\theta, \tau) + logP(\theta) \\ &= log\binom{R}{x_1, \ldots, x_M} + (1 - R)log(1 + R\tau) - \sum_{i=1}^{N} \alpha log\theta_i \\ &\quad + \sum_{j=1}^{M} logP(r_j) + \sum_{j=1}^{M} ((x_j - 1)log(P(r_j) + \tau x_j)) \end{aligned} \tag{3.9}$$

An EM algorithm can be used to maximize the above likelihood function. We first introduce a latent binary variable $z_{i,j}$ to indicate whether a read r_j comes from isoform t^i; *i.e.*, $z_{i,j} = 1$ if r_j comes from isoform t^i, and 0 otherwise. The joint log-likelihood function can be rewritten as (See Section 3.6.1 for the derivation):

$$\begin{aligned} & \log P(R, Z, \theta, \tau) \\ = & \sum_{j=1}^{M} \sum_{i=1}^{N} x_j z_{i,j} \log \theta_i - \sum_{i=1}^{N} \alpha \log \theta_i \\ + & \sum_{j=1}^{M} \sum_{i=1}^{N} (x_j - 1) z_{i,j} \log(P(r_j|t^i) + x_j \tau) \\ + & (1 - R) \log(1 + R\tau) + \sum_{j=1}^{M} \sum_{i=1}^{N} z_{i,j} \log P(r_j|t^i) + C \end{aligned} \tag{3.10}$$

where C is a constant independent of θ and τ.

In the E step of the EM algorithm, the expectation of $z_{i,j}$, $\gamma_{i,j}$, is evaluated

using the values of θ_i, τ and $P(r_j|t^i)$ as follows:

$$\gamma_{i,j} = \frac{\theta_i(P(r_j|t^i) + x_j\tau)}{\sum_{k=1}^{N} \theta_k P(r_k|t^i) + x_j\tau} \tag{3.11}$$

And in the M step, by maximizing Equation (3.10) with respect to the constraint $\sum_{i=1}^{N} \theta_i = 1$, θ_i is updated as:

$$\theta_i = \frac{N_i - \alpha}{\sum_{k=1}^{N} N_k - \alpha} \tag{3.12}$$

where

$$N_i = \sum_{j=1}^{M} x_j \gamma_{i,j} \tag{3.13}$$

The MLE value of τ has no close-form solution. Instead, by taking the derivation of Equation (3.10) *w.r.t.* τ, we get the following equation:

$$f(\tau) = \sum_{j=1}^{M} \sum_{i=1}^{N} \frac{\gamma_{i,j} x_j (x_j - 1)}{P(r_j|t^i) + x_j\tau} = \frac{R(R-1)}{1 + R\tau} \tag{3.14}$$

We use the Newton-Raphson method [102] to calculate the value of τ satisfying $f(\tau) = 0$ as follows. The derivative of Equation (3.14) is:

$$f'(\tau) = -\sum_{j=1}^{M} \sum_{i=1}^{N} \frac{\gamma_{i,j} x_j^2 (x_j - 1)}{(P(r_j|t^i) + x_j\tau)^2} + \frac{R^2(R-1)}{(1 + R\tau)^2} \tag{3.15}$$

and we iteratively update the value of τ as

$$\tau^{t+1} = \tau^t - \frac{f(\tau^t)}{f'(\tau^t)} \tag{3.16}$$

where τ^t is the value of τ at the tth iteration.

A *component elimination* EM algorithm [19, 4] can be used to find solutions that favor a small number of highly expressed isoforms. Compared with the standard EM algorithm, it applies an additional component elimination step to exclude components with small probabilities. This method is able to determine the number of components automatically without having to invoke any model selection criteria such as the *Bayesian Inference Criteria (BIC)*, *Minimum Message Length (MML)* principle, *etc.* During the EM iterations, a component elimination step eliminates isoform t^i if $N_i < \alpha$ (or set $\theta_i = 0$). Here, the negative Dirichlet parameter α can be interpreted as the minimum number of reads required for each isoform to proceed to the next iteration. In this component elimination EM, θ_i is fixed to 0 once its value reaches below 0 in Equation (3.12):

$$\theta_i = \frac{max(0, N_i - \alpha)}{\sum_{j=1}^{N} max(0, N_j - \alpha)} \tag{3.17}$$

However, in some component elimination steps (especially at the beginning of the EM iterations), there could be too many (or all) components satisfying the elimination condition $N_i < \alpha$. This is because the probability of each component is initialized randomly, which could be very small even for highly expressed isoforms if the number of components is large. Deleting all of them in one iteration may lead to a poor choice of components. As a result, we eliminate only one component with the minimum value of $N_i - \alpha$ in each iteration.

The parameter α controls the number of isoforms to be output. The higher the value of α is, the fewer isoforms are reported. Different values of α could be used for genes with different numbers of mapped reads. Based on our empirical experience from simulation tests (see the Results section below), we set $\alpha = \max\{10, 0.01R\}$ for a

gene with R mapped reads in our experiments.

3.4 Experimental results

We have implemented the above component elimination EM algorithm called *CEM* in C++. In this section, we test the algorithm on both simulated and real RNA-Seq data and compare its performance with two state-of-the-art algorithms for transcriptome assembly and isoform abundance level estimation that do not consider RNA-Seq biases (*i.e.*, IsoLasso [47] and Cufflinks [87]), and a recent extension of Cufflinks that takes biases into account [70]. For convenience, we will refer to the last algorithm simply as "Cufflinks-bias". To our best knowledge, Cufflinks-bias is the only algorithm in the literature that considers RNA-Seq biases and is capable of assembling transcriptome. Note that although SLIDE [45] was published after IsoLasso and Cufflinks, we do not compare with it here because it was only tested on *Drosophila melanogaster* transcriptome in [45]. During the comparison study, the parameters of all programs are tuned empirically to achieve their best performance.

3.4.1 Simulation

We simulate biased RNA-Seq reads similar to the method described in Chapter 2. Known isoforms from the *mus musculus* (mm9) annotation database are first downloaded from the UCSC genome browser [22]. Each isoform is then assigned a random abundance value that follows approximately a log-normal distribution [74, 3]. Afterwards, different numbers of reads are generated from each isoform according to the assigned abundance. Sequencing errors and different positional biases are then simulated to generate the actual reads.

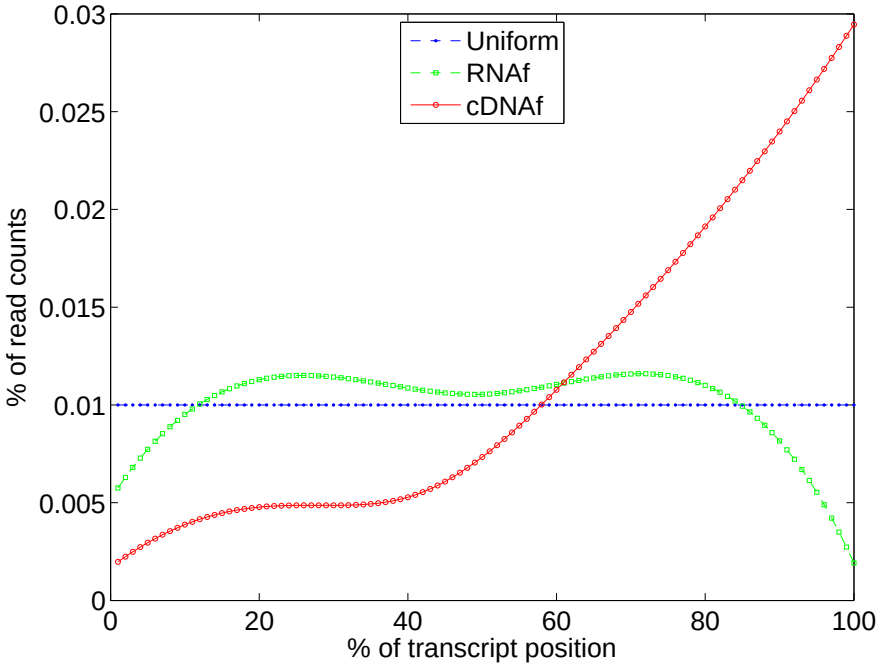


Figure 3.2: The positional bias models used in the simulation. Three different positional models are used, including the uniform model ("Uniform"), Illumina cDNA fragmentation model ("cDNAf") and Illumina RNA fragmentation model ("RNAf") [93, 35].

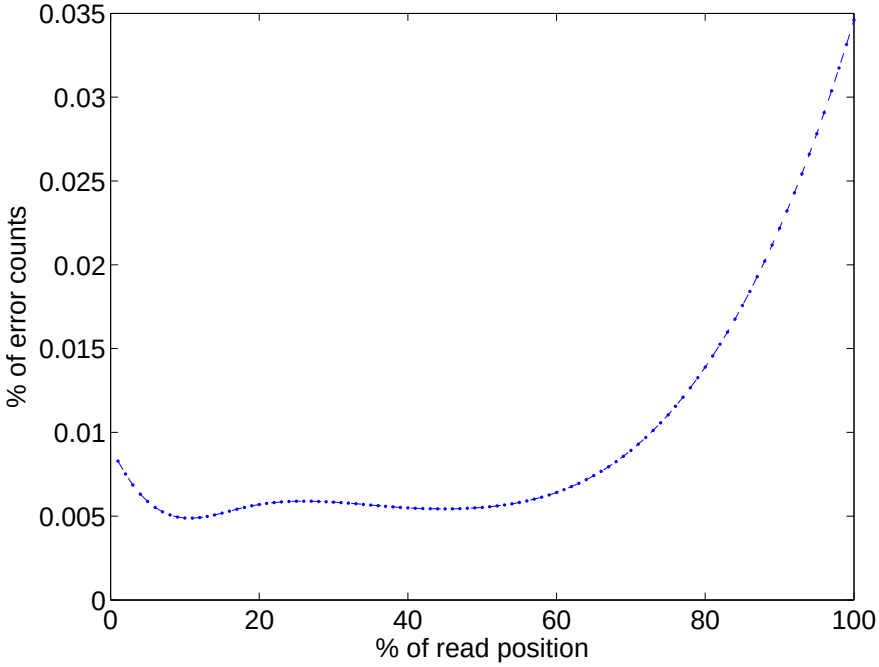


Figure 3.3: The sequencing error model used in the simulation. The sequencing error profile is summarized from real RNA-Seq data in [15].

During the simulation, three different positional profiles are provided to determine the position of each read, and 80 million single-end reads are generated for each profile, including the uniform positional model ("Uniform" for short) and two Illumina positional bias models (Figure 3.2). Both Illumina models reflect the positional biases caused by different fragmentation methods, including cDNA fragmentation (or "cDNAf") and RNA fragmentation (or "RNAf") [93, 35, 99]. We use the sequencing error profile in [15] to simulate sequencing errors (Figure 3.3), which is known to yield a non-uniform distribution in real RNA-Seq data: higher sequencing error is observed for positions at the end of a read [15].

The positional bias (*i.e.*, $f(k_{i,j})$ in Equation (3.6)) is learned from RNA-Seq data using a method similar to [99]. Basically, RNA-Seq reads are first mapped to the RefSeq transcript sequences [68] using Bowtie [41], where all possible mappings for each read are reported. A RefSeq sequence is selected to estimate its positional bias if the reads mapped to the sequence satisfy two conditions: (i) they cannot be mapped to other RefSeq sequences and (ii) the number of the reads is greater than 1000. The average of the positional biases in these sequences (about $2,000$) is then fed to the CEM algorithm. Figure 3.4 demonstrates that the estimated positional biases from different datasets are close to the real positional biases.

3.4.1.1 Performance on transcriptome assembly

The performance of transcriptome assembly results is evaluated in terms of both sensitivity and precision (Equation (2.10) and (2.11) in Chapter 2). To compare the effects of both positional and mappability biases on transcriptome assembly, we plot the sensitivity-precision curves of four programs: CEM, IsoLasso [47], Cufflinks [87] and

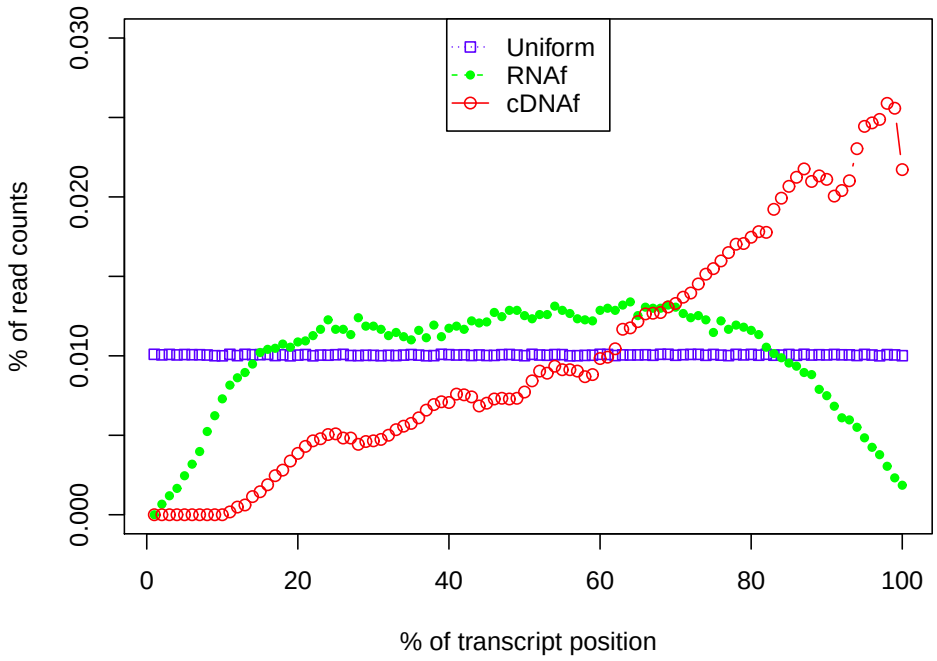


Figure 3.4: The estimated positional biases from simulated RNA-Seq data.

Cufflinks-bias [70].

Various values of sensitivity and precision in the curve are obtained by setting different abundance cutoffs used in the output of the programs. That is, for each cutoff, only predicted isoforms with estimated abundance levels higher than the cutoff value are output. Four different RNA-Seq reads are provided to the programs: reads with Uniform/cDNAf positional distributions and reads with/without mapping. For "reads without mapping" (or "w/o mapping" for short), the exact locations of the reads on the reference genome are provided; otherwise ("reads with mapping" or "mapping" for short), simulated reads are mapped to the reference genome using Tophat [86] to obtain the locations of the reads. Compared to the "reads with mapping" case, "reads without mapping" serves as an ideal dataset which is not affected by mappability biases.

Figure 3.5 compares the curves of both CEM and Cufflinks-bias. When pro-

vided with the correct mapping information (*i.e.*, w/o mapping), CEM and Cufflinks-bias both achieve high sensitivity (> 0.45) and precision (> 0.6). A high abundance cutoff allows only a small number of highly expressed isoforms to be retained. These isoforms are more likely to be correct (than those lowly expressed ones), which leads to low sensitivity and high precision for both CEM and Cufflinks-bias (except in groups 3 and 4 due to reasons explained below) in Figure 3.5. Compared with Cufflinks-bias, CEM achieves a better precision for the same level of sensitivity. CEM also performs best among all four algorithms, as seen in Figure 3.6 which shows the curves of all four algorithms using the cDNAf positional bias profile and "reads with mapping" option.

Both non-uniform positional biases and inaccurate read mapping have negative impact on transcriptome assembly. Compared with non-uniform positional bias dataset, higher sensitivity and precision values are observed for data generated using the Uniform positional profile. Interestingly, positional biases mainly affect the inference of lowly and moderately expressed isoforms. This could be seen from the diminishing differences between the sensitivity-precision curves for data with the Uniform and cDNAf positional biases in Figure 3.5 (see groups 1,2 and the CEM curves in groups 3,4). The reason is that lowly expressed isoforms are less likely to have sufficient read coverage to be assembled completely, since their junctions are less likely to be fully covered by reads [18].

The values of sensitivity and precision decrease drastically when correct mapping is not guaranteed. Figure 3.5 shows a 10%-15% decrease in sensitivity and a more than 20% decrease in precision in groups 3,4 compared with groups 1,2. Both repeat sequences of the genome and sequencing errors account for the decreased sensitivity and precision. This shows that mappability biases have a more profound effect on transcriptome assembly than positional biases. Different from positional biases, mappability

biases affect both highly and lowly expressed isoforms.

Interestingly, Cufflinks-bias shows reduced performance in both sensitivity and precision in groups 3 and 4 on "reads with mapping" for high abundance cutoffs. By inspecting the isoforms predicted by Cufflinks-bias carefully, we found that Cufflinks-bias is highly sensitive to mapping errors. For example, when the abundance cutoff is set as high as 500 FPKM, about 60% of the isoforms predicted by Cufflinks-bias come from regions with incorrectly mapped reads. Reads from the junctions of isoforms located in other regions could be mapped to these regions by TopHat because these junctions share identical sequences with those regions. As a result, the predicted isoforms are short compared to the read length, and Cufflinks-bias would greatly over-estimate their abundance levels, since it uses a fragment length model that assumes short *fragments* (short DNA sequences after fragmentation and before sequencing) are rare [87]. A specific example is given in Figure 3.7 and some statistics are given in Figure 3.8. A similar observation of this behavior of Cufflinks is also reported recently in [43]. CEM is less affected by this issue because it makes no assumption about the distribution of fragment lengths.

3.4.1.2 Longer read length improves both sensitivity and precision

To investigate the effect of read length on transcriptome assembly, we generate 80 million simulated reads of various read lengths (from 32bp to 200bp) using the uniform positional model (*i.e.*, without positional biases), and compare both values of sensitivity and precision of two programs (CEM and Cufflinks) in Figure 3.9. Here, no aboundance cutoff is applied to the predictions of either program.

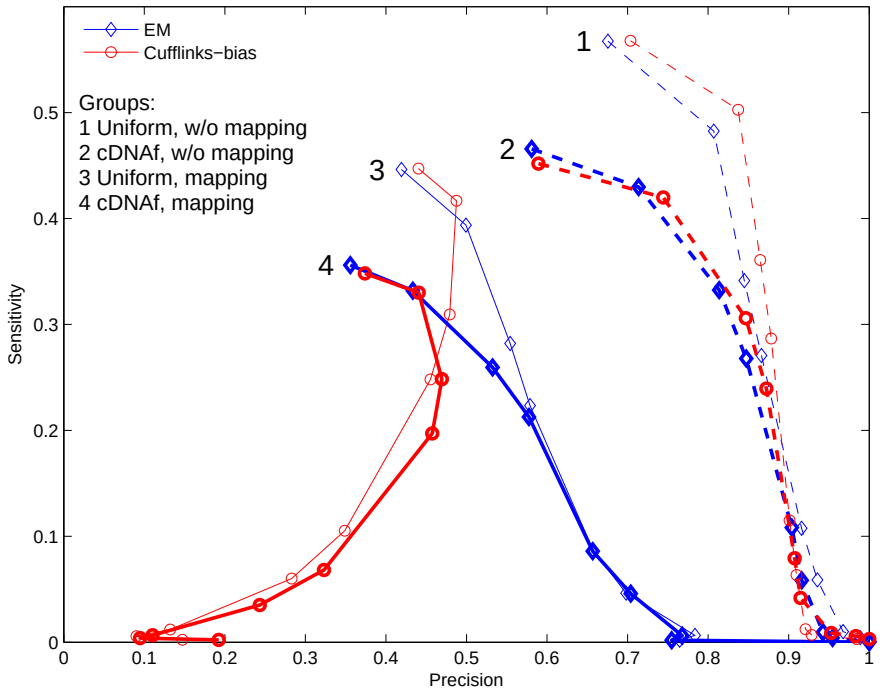


Figure 3.5: Sensitivity-precision curves of both CEM and Cufflinks-bias on four datasets. This figure shows the effect of both positional and mappability biases on CEM and Cufflinks-bias. "w/o mapping": correct read locations are provided; "mapping": reads are mapped to the reference genome using Tophat.

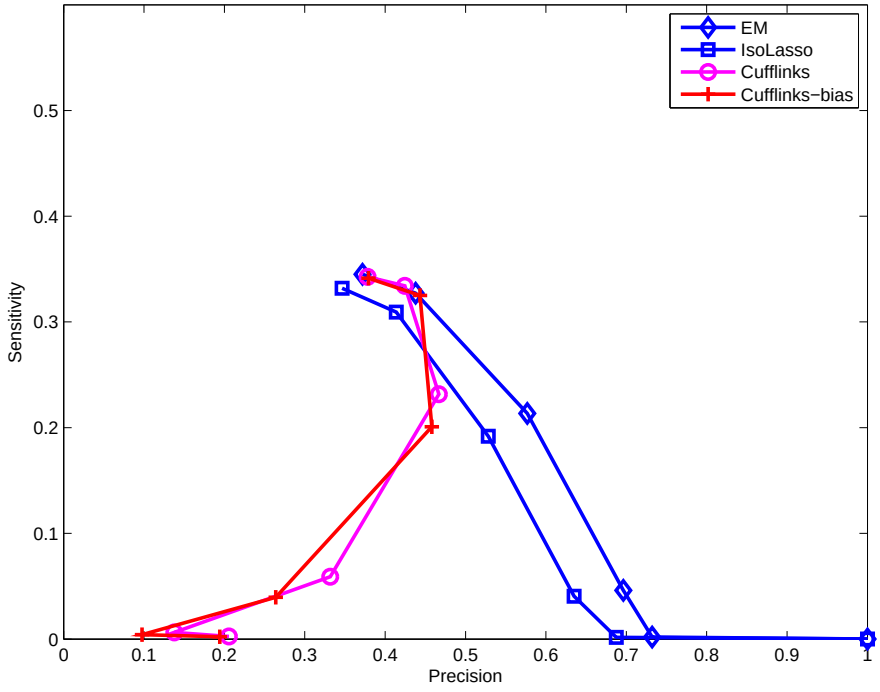


Figure 3.6: Sensitivity-precision curves of four different algorithms: CEM, IsoLasso, Cufflinks and Cufflinks-bias. Here, the curves for CEM and Cufflinks-bias correspond to those in group 4 of Figure 3.5.

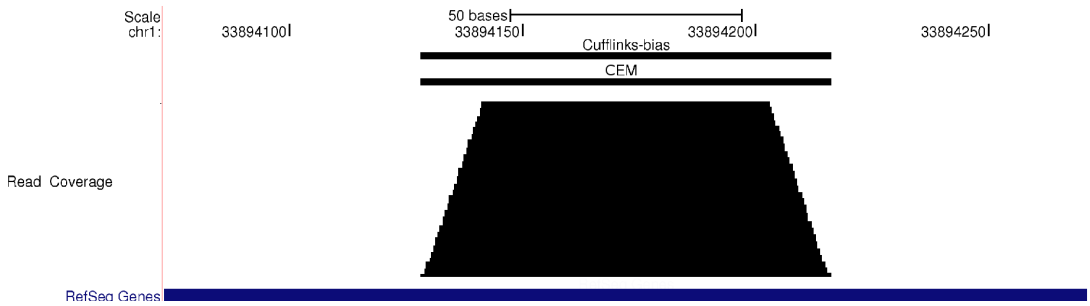


Figure 3.7: An example of incorrectly assembled transcript due to read mapping error. In this example, the (simulated) reads from some exon junctions of gene Rbm10 (on chromosome X:46,889,828-46,891,700) are mapped to chromosome 1 due to sequence similarity. Both Cufflinks-bias and CEM make incorrect predictions. However, the predicted FPKM by Cufflinks-bias is much higher (62094.1) than that by CEM (119.5).

Figure 3.9 shows that both sensitivity and precision increase as longer reads are used for assembly. However, such improvements tend to slow down as reads get longer. For example, when read length increases from 32bp to 50bp, the sensitivity of CEM (on reads with mapping) increases from 0.35 to 0.41. This increase is much more drastic than the improvement obtained when increasing read length from 100bp to 200bp, which is only around 0.03. Similar trends can be observed for Cufflinks (on reads with mapping) and for the value of precision.

Longer reads incur less ambiguity in mapping to the reference genome. For example, among all 32bp reads mapped to the reference genome, 25% can be mapped to multiple locations. But for the 200bp reads, only 12% are mapped to more than one location in the reference genome. However, inspite of the reduced ambiguity in mapping when longer reads are used, the differences in sensitivity and precision of both programs on reads with/without mapping are consistently observed in Figure 3.9 (which is always about 0.1). This shows that even for long reads, the mappability bias still affects the accuracy of transcriptome assembly.

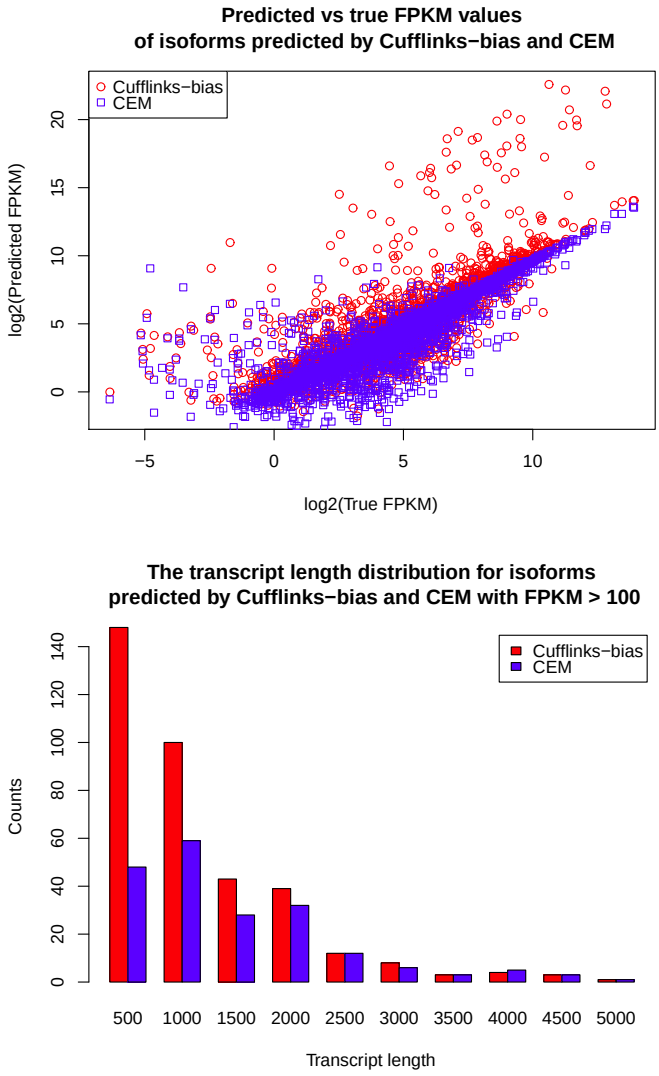
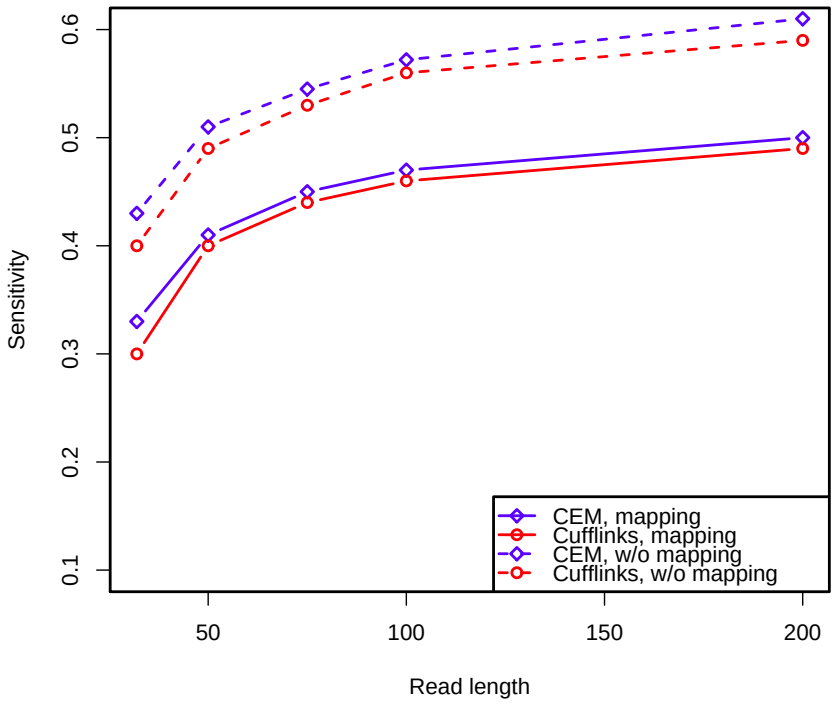


Figure 3.8: Detailed analysis of highly expressed isoforms predicted by Cufflinks-bias and CEM on the simulated data. The first plot compares the predicted and true FPKM values of the isoforms predicted by Cufflinks-bias and CEM, where some isoforms have their FPKM values greatly over-estimated by Cufflinks-bias (the red circles above the main diagonal). Although many of these isoforms are short (and false), they cannot be easily removed by using a simple length threshold, since many of the predicted isoforms are short and many of them are true. The second plot shows that for highly expressed isoforms (FPKM>100), the proportion of short transcripts (<500) of Cufflinks is much higher than CEM.



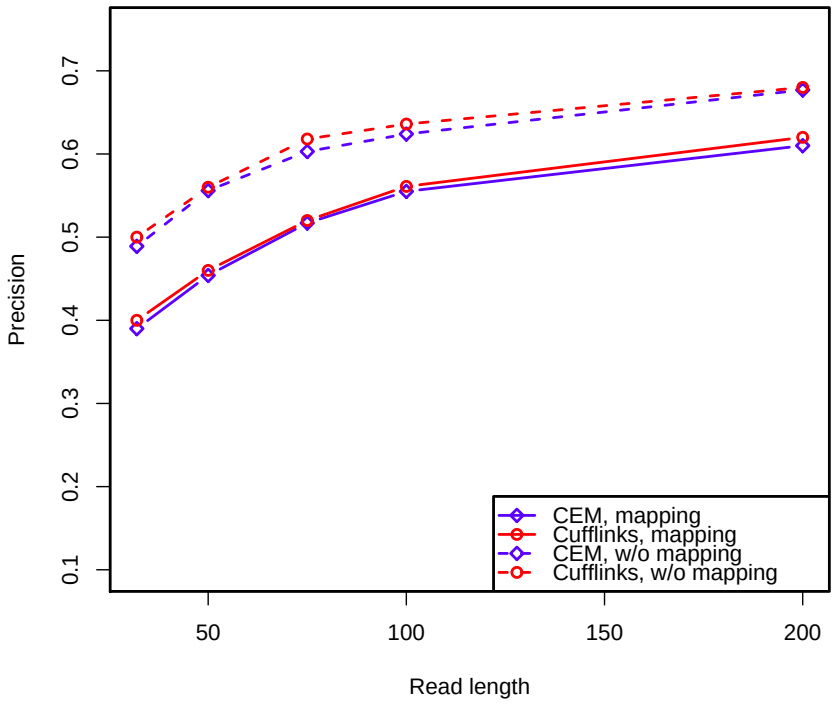


Figure 3.9: The effect of read length on both sensitivity (top) and precision (bottom).

Table 3.1: Comparison of the R^2 values of the four algorithms in isoform abundance estimation on data with various positional biases.

Dataset	CEM	IsoLasso	Cufflinks	Cufflinks-bias
Uniform	**0.90**	0.87	0.86	0.89
RNAf	**0.87**	0.80	0.83	0.84
cDNAf	**0.84**	0.72	0.76	0.82

3.4.1.3 Performance on abundance level estimation

We assemble transcripts from simulated RNA-Seq reads using the above algorithms, and then match their results to known mouse isoforms. For the matched isoforms, we compare the logarithms of the predicted and true abundance levels in Table 3.1 using *coefficient of determination* (*i.e.*, the R^2 value).

As shown in the table, all algorithms achieve high precision in abundance estimation on both Uniform and RNAf datasets ($R^2 > 0.8$), but CEM outperforms the other three methods on all datasets. The cDNAf positional profile contains more biases compared with the RNAf profile, since it has a more extreme head and tail positional distribution as shown in Figure 3.2. Not surprisingly, lower R^2 value is obtained for all methods on data with cDNAf positional biases. On the other hand, Cufflinks-bias demonstrates a clear advantage over Cufflinks on data with cDNAf positional biases.

3.4.2 Real data analysis

3.4.2.1 Correlation with MAQC data

We compare the abundance estimations for the Microarray Quality Control (MAQC) [50] Human Brain Reference (HBR) sample between Taqman qRT-PCR measurements and RNA-Seq analysis. The RNA-Seq reads and qRT-PCR measurements are downloaded from the NCBI SRA archive (accession number SRA012427) and Gene Expression Omnibus (accession number GSE5350), respectively. RNA-Seq reads are first

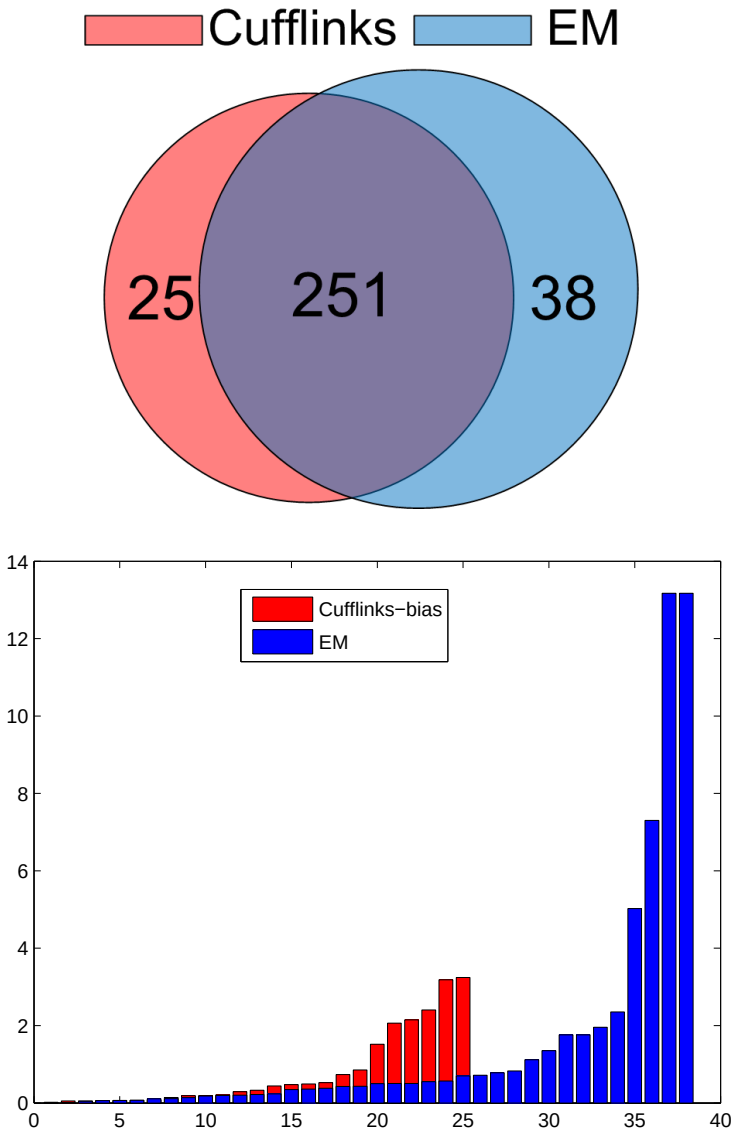


Figure 3.10: Comparison of the transcriptome assembly results between CEM and Cufflinks-bias. Top: The assembled transcripts by CEM and Cufflinks-bias match 289 and 276 of the 1097 Taqman qRT-PCR transcripts, respectively. Bottom: The distributions of the qRT-PCR measurements of the 38 and 25 transcripts uniquely assembled by CEM and Cufflinks.

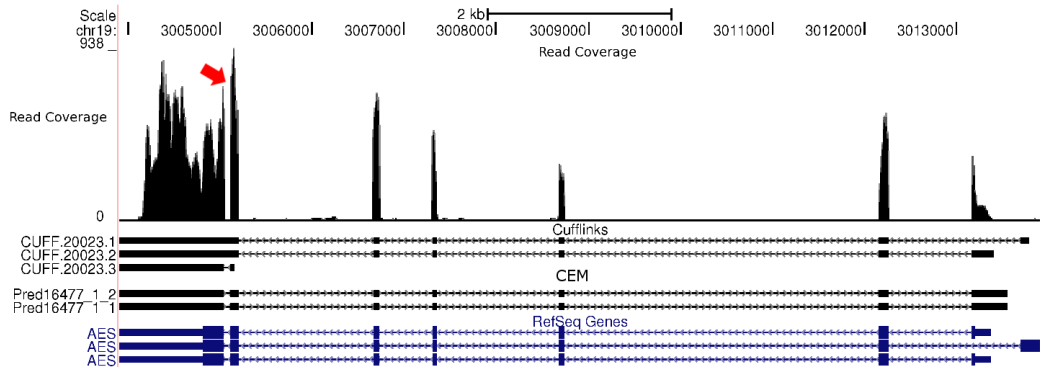


Figure 3.11: An example of highly expressed isoforms identified by CEM, but not by Cufflinks-bias. CEM correctly assembles two of the three known isoforms of the AES gene, but Cufflinks-bias misses the first short intron (the red arrow) in its first two predictions and thus fails to detect all three isoforms.

assembled into transcripts, whose abundance estimations (in RPKM or FPKM) are then correlated with the Taqman qRT-PCR measurements. Since Taqman qRT-PCR only measures the expression levels of genes, we only compare gene abundance estimations. Given the isoform abundance levels estimated from the RNA-Seq data, the expression level of a gene is obtained by summing up the abundance levels of all isoforms induced by the gene.

Among 1097 Taqman qRT-PCR measurements of genes, 289 and 276 are correctly assembled by CEM and Cufflinks-bias, respectively, as shown in Figure 3.10 top. The intersection of both programs covers 251 transcripts ($> 85\%$), showing a high consistency between both methods. CEM recovers slightly more (13) transcripts than Cufflinks-bias. For the 38 and 25 transcripts uniquely assembled by CEM and Cufflinks-bias, Figure 3.10 bottom plots the distribution of their qRT-PCR measurements. A few highly expressed transcripts are correctly assembled by CEM but not by Cufflinks-bias. An interesting example is the three isoforms from the AES gene on chromosome 19 (see Figure 3.11). CEM correctly assembles two of them, but Cufflinks-bias assembles none of them correctly because it failed to recognize the first short intron of the gene.

To compare the abundance estimations, we run the four algorithms in two different ways. In the first case (the "*de novo*" approach), the algorithms are invoked to assemble transcripts and the results are matched against the known structures of RefSeq transcripts corresponding to Taqman gene measurements (note that here the term "*de novo*" has a different meaning than it does in "*de novo* assembly"). For the matched genes, the abundance levels estimated by the four algorithms are compared to qRT-PCR measurements. In the second case (the "refonly" approach), the structures of all Taqman transcripts are provided, and their estimated abundance levels are compared to the Taqman qRT-PCR measurements.

Figure 3.12 shows the R^2 values between RNA-Seq and Taqman qRT-PCR measurements for both "*de novo*" and "refonly" approaches. We first compare the R^2 values of the top 100 predicted highly expressed genes in Figure 3.12 top (note that different number of genes between 50 and 200 give similar results). For these genes, the "*de novo*" approach shows higher values of R^2 than the "refonly" approach. And among the four compared methods, CEM achieves the highest correlation.

However, when correlating RNA-Seq based abundance estimations of all genes (instead of only highly expressed genes) to the Taqman qRT-PCR measurements, Cufflinks-bias shows a clear improvement over Cufflinks as shown in Figure 3.12 bottom, and achieves the highest correlation among all 4 algorithms. This increased performance of Cufflinks-bias suggests that Cufflinks-bias corrects biases the best in the estimation of the abundance levels of moderately and lowly expressed genes, while CEM algorithm works the best for highly expressed genes. This is consistent with the observed advantage of CEM in the simulated data experiments.

The expression levels of genes also have an impact on the performance of the "*de novo*" and "refonly" approaches. For the 100 genes with the highest predicted expression

levels, the R^2 values from the "*de novo*" approach are higher than those obtained by "refonly" approach where known transcript structures are provided. However, for all genes, a large improvement in R^2 values is observed for the "refonly" approach over the "*de novo*" approach as shown in Figure 3.12 bottom. This suggests that knowing correct transcript structures is crucial for estimating the expression levels of lowly and moderately expressed genes, since it might be difficult for the algorithms to correctly assemble the isoforms of these genes from RNA-Seq reads.

3.4.2.2 Regression slope comparison

We also analyze the regression coefficient between RNA-Seq and Taqman qRT-PCR measurements in Table 3.2. The regression slope reflects the fold change between the expression levels of genes, where the ideal slope of 1.0 indicates that two methods are perfectly consistent in detecting fold changes between genes. Table 3.2 shows that although the R^2 values are relatively low for the top 100 genes using the "refonly" approach, CEM, IsoLasso and Cufflinks are able to detect fold changes quite accurately (slope > 0.9). On the other hand, Cufflinks-bias is unable to match the performance on these highly expressed genes (slope $= 0.75$) for some reason (*e.g.*, perhaps due to incorrect correction of their expression levels). Table 3.2 also shows that providing transcript structures helps the fold change detection (the slopes in the *refonly* columns are above 0.75 while the slopes in the "*de novo*" columns are only between 0.4 and 0.5).

3.4.2.3 Running time comparison

Figure 3.13 compares the running times of all four algorithms for processing 80M paired-end reads on a Linux machine with 16G memory and 2.6GHz 8-core CPU.

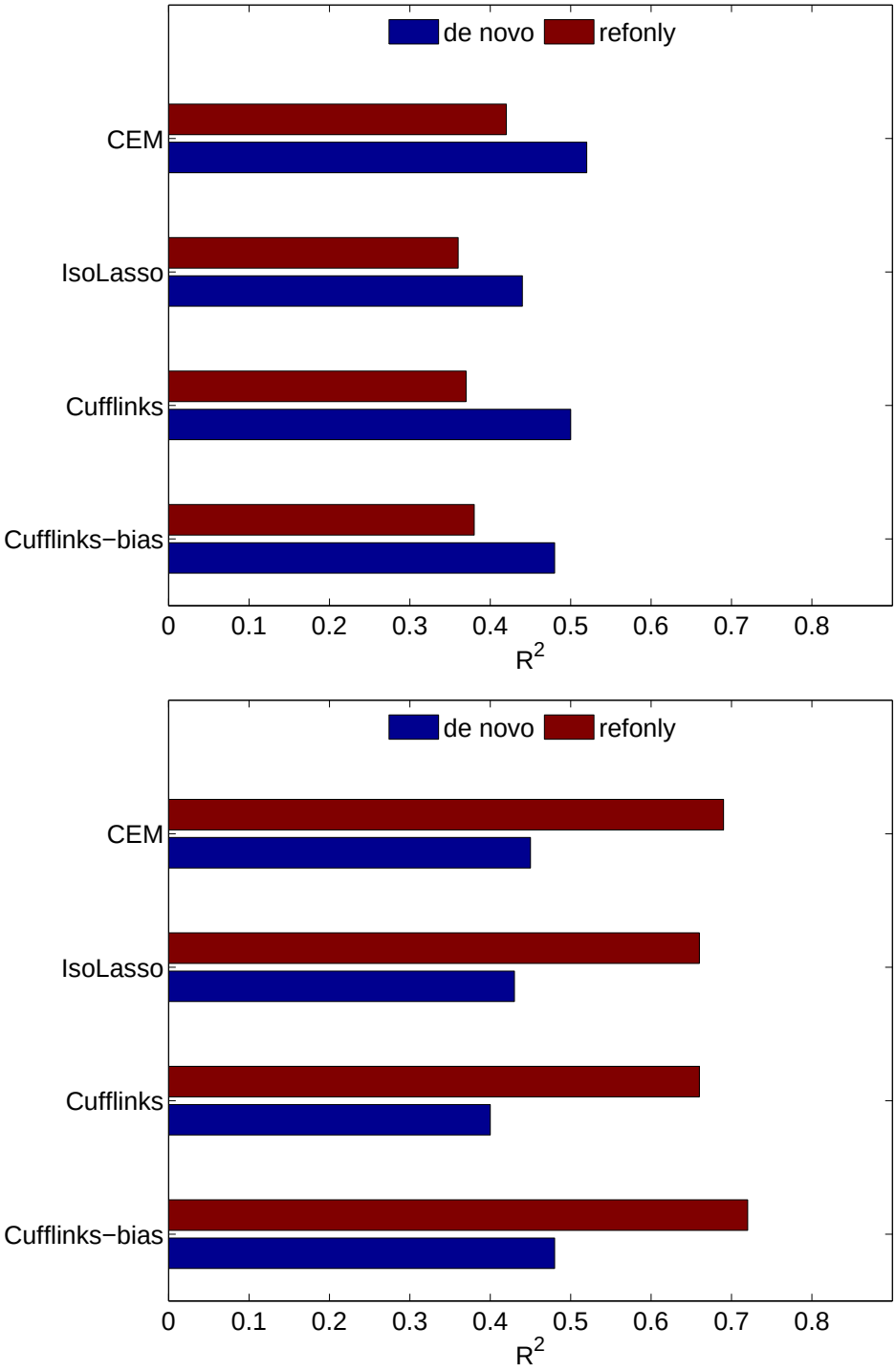


Figure 3.12: The R^2 values between RNA-Seq and Taqman qRT-PCR measurements of the MAQC HBR sample for the top 100 predicted highly expressed genes (top) and for all genes (bottom). Red and blue bars in both figures show the R^2 values when transcript structures are provided to the algorithms (*i.e.*, the "refonly" approach), or by assembly.

Table 3.2: The regression lines between RNA-Seq (y) and Taqman qRT-PCR measurements (x) in log scale.

Algorithm	refonly top 100	*de novo* top 100	refonly	*de novo*
CEM	$0.97x+5.8$	$0.42x+3.8$	$0.71x+5.8$	$0.54x+3.4$
IsoLasso	$0.90x+5.7$	$0.40x+3.7$	$0.72x+5.9$	$0.53x+3.3$
Cufflinks	$0.92x+3.1$	$0.43x+3.0$	$0.72x+3.2$	$0.53x+3.3$
Cufflinks-bias	$0.76x+2.0$	$0.43x+3.5$	$0.73x+3.0$	$0.55x+3.8$

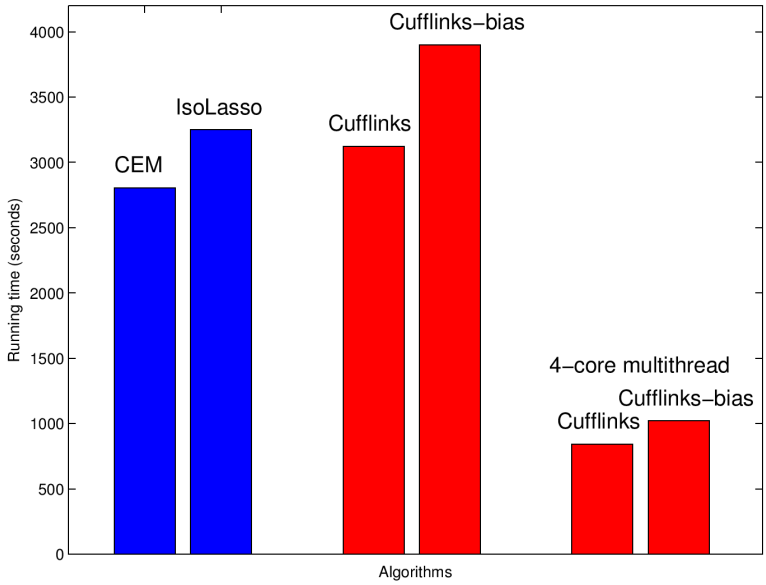


Figure 3.13: The running times of CEM, IsoLasso, Cufflinks, and Cufflinks-bias using 80M paired-end reads. Both Cufflinks and Cufflinks-bias can use multiple threads.

Since both Cufflinks and Cufflinks-bias provide the option of multithreading ("-p" option), we also include the results of both Cufflinks and Cufflinks-bias using 4 threads. We can see that for both Cufflinks and Cufflinks-bias, using multiple threads greatly reduces the processing time needed: only about 1/4 of the time is required for 4 threads compared with 1 thread. This is partly because transcriptome assembly can be trivially performed in parallel, allowing reads mapped to different genes to be processed simultaneously. If only a single thread is used, the speeds of all algorithms are approximately at the same level, with CEM slightly leading the edge.

3.4.2.4 Exon inclusion ratio analysis

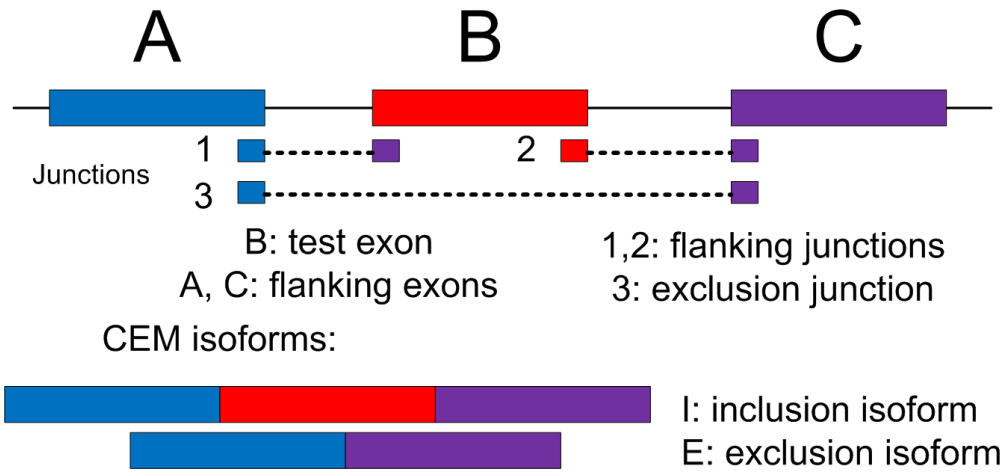


Figure 3.14: The exon inclusion ratio (Ψ) calculation model. Both the "direct" and isoform abundance based models are used to calculate the exon inclusion ratio (Ψ) value of test exon B. The "direct" method calculates both the inclusion density (d_I, the read density of exon B and junction 1,2) and exclusion density (d_E, the read density of junction 3). The abundance based method calculates the abundance levels of both inclusion isoform (containing exons A, B and C) and exclusion isoform (containing exons A and C) using CEM.

The *exon inclusion ratio* (or *"percent spliced in"* value, Ψ) measures the percentage of mRNA transcripts that include an exon to the total amount of transcripts that include or exclude that exon. The Ψ value is frequently used to study the mechanism of

Table 3.3: The R^2 values and regression coefficients between the exon inclusion values calculated by RNA-Seq and qRT-PCR analyses.

Dataset	CTRL		KD	
Metric	direct	CEM	direct	CEM
R^2	0.81	0.86	0.84	0.93
Regression	$0.64x$+0.3	$0.80x$+0.5	$0.84x$+0.1	$0.94x$+0.1

alternative splicing [62, 100, 90]. In [100], both qRT-PCR and mRNA-Seq experiments are performed to measure the Ψ values for human HEK 293T cells, including hnRNP H knockdown (or "KD") cells and corresponding control (or "CTRL") cells. RNA-Seq reads are first mapped to the human reference genome, and the Ψ value is calculated as

$$\Psi = d_I/(d_I + d_E) \tag{3.18}$$

where d_I (*inclusion density*) is defined as the read density of the test exon and its two flanking junctions, and d_E (*exclusion density*) is the read density of the exclusion junction formed by the two flanking exons (see Figure 3.14). However, this method (called the "direct" method) is sensitive to the value of d_E, which may not be accurate if few reads are mapped to the exclusion junction. Alternatively, the Ψ value can be calculated based on the abundance levels of two isoforms including and excluding the test exon:

$$\Psi = q_I/(q_I + q_E) \tag{3.19}$$

where q_I and q_E are the estimated abundance levels of two isoforms including and excluding the test exon, respectively, as illustrated in Figure 3.14. We calculate the Ψ values using both the "direct" method and the above method based on isoform abundance levels estimated by our CEM algorithm, and correlate the results with the Ψ values calculated by qRT-PCR experiments in Table 3.3.

Table 3.3 shows a significantly improved correlation using the isoform abundance method based on CEM over the "direct" method. The CEM algorithm achieves higher PCC and R^2 values on both CTRL and KD datasets, and the regression slope (> 0.9) on the KD dataset shows that the Ψ values obtained by CEM are quite consistent with the qRT-PCR data.

3.5 Conclusion

Biases in RNA-Seq data are difficult to deal with because they affect both transcriptome assembly and isoform abundance estimation. The current literature focuses on correcting biases for isoform abundance estimation, but little has been done for transcriptome assembly. In this paper, we present a quasi-multinomial distribution based statistical framework and component elimination EM algorithm for both transcriptome assembly and isoform abundance estimation from biased RNA-Seq data. Biases are captured by a single parameter τ in the quasi-multinomial model, and the component elimination EM algorithm ensures that good interpretation (or sparsity) is achieved in transcriptome assembly.

Both simulated and real data experiments reveal interesting effects of different biases. Although the precision and sensitivity of a method in transcriptome assembly are affected by both positional and mappability biases, the recovery of isoforms/genes with different abundance levels are affected differently. While mappability biases reduce the sensitivity and precision for all genes, positional biases have a negative effect mainly on lowly or moderately expressed genes. A comparison between our CEM algorithm and the other methods in the literature shows that for highly expressed isoforms, our algorithm achieves higher sensitivity and precision in assembly and higher accuracy in

isoform abundance estimation.

3.6 Additional remarks

3.6.1 Derivation of the joint log-likelihood function

The derivation of Equation (3.10) is as follows.

From Equations (3.6) and (3.7), we have

$$P(r_j) = \sum_{i=1}^{N} \theta_i P(r_j|t^i) \tag{3.20}$$

$$P(r_j) + \tau x_j = \sum_{i=1}^{N} \theta_i \left(P(r_j|t^i) + \tau x_j \right) \tag{3.21}$$

since $\sum_{i=1}^{N} \theta_i = 1$. By using the latent binary variable $z_{i,j}$ introduced in Section 3.3.2, both equations can be re-written as

$$P(r_j) = \prod_{i=1}^{N} \left(\theta_i P(r_j|t^i) \right)^{z_{i,j}} \tag{3.22}$$

$$P(r_j) + \tau x_j = \prod_{i=1}^{N} \left(\theta_i (P(r_j|t^i) + \tau x_j) \right)^{z_{i,j}} \tag{3.23}$$

The joint log-likelihood function in Equation (3.10) is the logarithm of the product of Equations (3.7) and (3.8). Using both Equations (3.22) and (3.23), we

obtain

$$
\begin{aligned}
logP(R,Z,\theta,\tau) &= logP(R|\theta,\tau) + logP(\theta) \\
&= log\binom{R}{x_1,\ldots,x_M} + (1-R)log(1+R\tau) - \sum_{i=1}^{N} \alpha log\theta_i \\
&\quad + \sum_{j=1}^{M} logP(r_j) + \sum_{j=1}^{M} ((x_j-1)log(P(r_j)+\tau x_j)) \\
&= \sum_{j=1}^{M}\sum_{i=1}^{N} z_{i,j}(log\theta_i + logP(r_j|t^i)) \\
&\quad + \sum_{j=1}^{M}\sum_{i=1}^{N} (x_j-1)z_{i,j}(log\theta_i + log(P(r_j|t^i)+\tau x_j)) \\
&\quad + (1-R)log(1+R\tau) - \sum_{i=1}^{N} \alpha log\theta_i + C \\
&= \sum_{j=1}^{M}\sum_{i=1}^{N} x_j z_{i,j} \log\theta_i - \sum_{i=1}^{N} \alpha \log\theta_i \\
&+ \sum_{j=1}^{M}\sum_{i=1}^{N} (x_j-1)z_{i,j} \log(P(r_j|t^i)+x_j\tau) + (1-R)\log(1+R\tau) \\
&+ \sum_{j=1}^{M}\sum_{i=1}^{N} z_{i,j} \log P(r_j|t^i) + C
\end{aligned} \tag{3.24}
$$

3.6.2 Quasi-multinomial distributions, quasi-binomial distributions and generalized Poisson distributions

Generalized Poisson (GP) distributions have been used to correct biases at the gene level [79]. Here, we show that maximizing the log-probability objective function in Equation (3.7) with respect to τ is equivalent to maximizing the objective function using GP distributions with respect to ρ.

For simplicity, consider a gene with a single isoform (*i.e.*, $N = 1$). Take read counts at individual positions into consideration, then $\gamma_{i,j} = 1$ and $P(r_j|t^i) = P(r_j) =$

$1/M$, and Equation (3.14) becomes

$$f(\tau) = \sum_{j=1}^{M} \frac{x_j(x_j - 1)}{P(r_j) + x_j\tau} - \frac{R(R-1)}{1 + R\tau} \tag{3.25}$$

When $R \to \infty$, $P(r_j) \to 0$, $\tau \to 0$, $RP(r_j) \to \upsilon$, and $R\tau \to \beta$, where υ and β are some constants, the quasi-multinomial distribution (Equation (3.7)) approaches the product of $M - 1$ GP distributions $GP(\lambda, \rho) = GP(\upsilon/(1 + \beta), \beta/(1 + \beta))$ conditioned on R being fixed [11]. Now we show that under these conditions, maximizing the log-probability objective function in Equation (3.7) with respect to τ is equivalent to maximizing the joint log-probability of M GP distributions with respect to ρ (here we need M GP distributions instead of $M - 1$ because R is not fixed).

The GP distribution with parameter (λ, ρ) is [79]:

$$P(X = x) = \lambda(\lambda + x\rho)^{x-1} e^{-\lambda - x\rho}/x! \tag{3.26}$$

If we observe M read counts $x_1, \ldots, x_M$ at M positions, the joint log-probability is

$$\sum_{i=1}^{M} \log P(X_i = x_i) = \sum_{i=1}^{M} (x_i - 1)\log(\lambda + x_i\rho) - x_i\rho + M(\log\lambda - \lambda) - \sum_{i=1}^{M} \log x_i! \tag{3.27}$$

Setting the derivative of Equation (3.27) to zero, we have

$$\sum_{i=1}^{M} \frac{(x_i - 1)x_i}{\lambda + x_i\rho} - \sum_{i=1}^{M} x_i = 0 \tag{3.28}$$

Let $R = \sum_{i=1}^{M} x_i$, then $\hat{x} = R/M$, where $\hat{x}$ denotes the mean value of an x_i. If we replace λ with the Maximum Likelihood Estimation (MLE) value $\hat{\lambda} = \hat{x}(1 - \rho)$ [79],

then the MLE value of ρ should satisfy

$$\sum_{i=1}^{M} \frac{x_i(x_i-1)}{\hat{x}+(x_i-\hat{x})\rho} - M\hat{x} = 0 \tag{3.29}$$

which is equivalent to the Newton-Raphson update equation in [79]. [2]

For $R \to \infty$, $P(r_j) \to 0$, $\tau \to 0$, $(R-1)P(r_j) \to \upsilon$, and $\frac{(R-1)\tau}{1+R\tau} \to \rho$, we obtain $\lambda = \upsilon/(1+\beta) \to (R-1)/M(1+R\tau)$ and $\rho = \beta/(1+\beta) \to (R-1)\tau/(1+R\tau)$. Substituting them into Equation (3.28), we have

$$\sum_{i=1}^{M} \frac{(x_i-1)x_i}{\frac{1}{M}\frac{R-1}{1+R\tau} + x_i\frac{R-1\tau}{1+R\tau}} - R = 0 \tag{3.30}$$

Simplifying the equation, we obtain

$$\sum_{i=1}^{M} \frac{(x_i-1)x_i}{P(r_j)+x_i\tau} = \frac{R(R-1)}{1+R\tau}, \tag{3.31}$$

which is the same as Equation (3.25).

Here we show that the quasi-multinomial distribution in Equation (3.7) in fact reduces to a quasi-binomial distribution of type II when $M = 2$ [10, 34]. If $M = 2$, write $P(r_1) = p_1$, $P(r_2) = p_2$ and $R = x_1 + x_2$, where $p_1 + p_2 = 1$. Rewrite Equation (3.7) as follows:

$$P(R|\theta,\tau) = \binom{R}{x_1, x_2} p_2 \frac{p_1}{1+R\tau}\left[\frac{p_1}{1+R\tau} + \frac{\tau}{1+R\tau}x_1\right]^{x_1-1}\left[1 - \frac{p_1}{1+R\tau} - \frac{\tau}{1+R\tau}x_1\right]^{R-x_1-1} \tag{3.32}$$

[2] Notice that in [79], there is a typo in the Newton-Raphson equation, and the correct equation should read:

$$\sum_{i=1}^{n} \frac{x_i(x_i-1)}{\hat{x}+(x_i-\hat{x})\lambda} - n\hat{x} = 0, \text{where } \hat{x} = \sum_{i=1}^{n}\frac{x_i}{n}$$

Let $p' = \frac{p_1}{1-R\tau}$ and $\alpha = \frac{\tau}{1+R\tau}$. Then

$$\frac{1-p'-R\alpha}{1-R\alpha} = \frac{1-\frac{p_1}{1+R\tau}-R\frac{\tau}{1+R\tau}}{1-R\frac{\tau}{1+R\tau}} = \frac{1+R\tau-p_1-R\tau}{1+R\tau-R\tau} = p_2 \tag{3.33}$$

Incorporating the above equation, Equation (3.32) becomes

$$P(R|\theta,\tau) = \binom{R}{x_1, x_2} \frac{(1-p'-R\alpha)p'}{1-R\alpha} (p+x_1\alpha)^{x_1-1} \left(1-p'-x_1\alpha\right)^{R-x_1-1} \tag{3.34}$$

which is a quasi-binomial distribution of type II (Equation (4) in [10]). [3]

3.6.3 Incorporating additional information into the quasi-multinomial model

As mentioned in Section 3.2, additional information can be incorporated into our quasi-multinomial model to improve the estimation of isoform abundance levels. Such information may include multi-reads [61, 44] (which is usually discarded), the distance between read pairs in paired-end reads [87], *etc.* Here we show how such information can be incorporated into our model. The quasi-multinomial distribution in Equation (3.7) can be re-written as:

$$P(R|\theta,\tau) = \binom{R}{x_1, \ldots, x_M} (1+R\tau) \prod_{j=1}^{M} \frac{P(r_j)}{1+R\tau} \left(\frac{P(r_j)+\tau x_j}{1+R\tau}\right)^{x_j-1} \tag{3.35}$$

[3] Notice that there is a typo in Equation (4) of [10], and the equation should be corrected as:

$$P(X=x) = \binom{n}{x} \frac{(1-p-n\alpha)p}{1-n\alpha} (p+x\alpha)^{x-1} (1-p-x\alpha)^{n-x-1}$$

Following the notations in [10], define $p = a(a+b+n\theta)^{-1}$ and $\alpha = \theta(a+b+n\theta)^{-1}$. Then

$$\frac{(1-p-n\alpha)p}{1-n\alpha} = \frac{a}{a+b+n\theta} \frac{1-\frac{a}{a+b+n\theta}-n\frac{\theta}{a+b+n\theta}}{1-n\frac{\theta}{a+b+n\theta}} = \frac{ab}{(a+b)(a+b+n\theta)}$$

Define $P_1(r_j) = P(r_j)/(1 + R\tau)$ and $P_2(r_j) = (P(r_j) + \tau x_j)/(1 + R\tau)$. Both $P_1(r_j)$ and $P_2(r_j)$ can be interpreted as the "adjusted" probability of all reads falling into exon j: one of the x_j mapped reads has the "adjusted" probability $P_1(r_j)$ and the rest $P_2(r_j)$. The probabilities of the reads are thus adjusted according to the number of reads mapped to exon j. Denote all the x_j reads mapped exon j as $r_{1,j}, \ldots, r_{x_j,j}$. We re-write Equation (3.35) as:

$$P(R|\theta, \tau) = \binom{R}{x_1, \ldots, x_M}(1 + R\tau)\prod_{j=1}^{M}\prod_{m=1}^{x_j} P'(r_{m,j}) \tag{3.36}$$

where $P'(r_{m,j})$ is the "adjusted" probability of the x_j reads mapped to exon j. One of the read is assigned probability $P'(r_{m,j}) = P_1(r_j)$ and the rest are assigned probability $P_2(r_j)$.

We can incorporate additional information such as multi-reads and the distance between read pairs in paired-end reads into the "adjusted" probability $P'(r_{m,j})$. For example, if the distance between read pairs $d_{m,j}$ of the paired-end read $r_{m,j}$ (also called its *span*) follows some probability distribution $g(d_{m,j})$, then

$$P(r_{m,j}) = \sum_{i=1}^{N} P(r_{m,j}|t^i)P(t^i) = \sum_{i=1}^{N} \theta_i f(k_{m,i})g(d_{m,j}) \tag{3.37}$$

where $k_{m,i}$ denotes the location $r_{m,j}$ in t^i and f the location distribution function (see Equation (3.6)).

3.6.4 Comparing CEM with NURD

NURD ([99]) is a software package for improving isoform expression level estimation based on non-uniform read distributions. Here, we compare the performance of

both CEM and NURD using three simulated RNA-Seq datasets mentioned in Section 3.4.1. Since NURD does not assemble transcripts, mm9 known isoforms are provided as the input to both programs (the same as the "refonly" approach in Section 3.4.2.1). Reads are mapped to the reference genome using Tophat in CEM and to the transcript sequences using Bowtie in NURD, as required by each program.

Table 3.4: **The R^2 values of CEM and NURD in isoform abundance estimation with various positional biases.**

Dataset	CEM	NURD
Uniform	0.89	0.85
RNAf	0.88	0.82
cDNAf	0.86	0.83

Table 3.4 compares the R^2 values of both programs in isoform abundance estimation using datasets with various positional biases. (Note that other biases also exist in the datasets.) The R^2 values of CEM are higher in all datasets, demonstrating that our model efficiently captures different biases in simulated datasets. Recall that NURD can only handle positional biases and is unable to correct sequencing or mappability biases.

3.6.5 The effect of different isoform abundance distributions

In our simulation experiments, the expression levels of isoforms are assigned random values that follow a log-normal distribution. This log-normal model assumes that only a few isoforms are highly expressed in a gene, and is a good approximation of the real isoform expression distributions ([1], [3]). We also tested the hypothesis that isoform expression levels follow a geometric distribution, and compared the effect of this hypothesis on transcriptome assembly and isoform expression level estimation using simulated RNA-Seq datasets. In these experiments, all procedures are identical to

Table 3.5: The effect of different isoform expression level distribution models on transcriptome assembly and isoform expression level estimation.

		Sensitivity	Precision	R^2
geometric	Cufflinks	0.57	0.56	0.53
	CEM	0.62	0.63	0.65
log-normal	Cufflinks	0.42	0.50	0.83
	CEM	0.44	0.52	0.85

the experiments described in Section 3.4.1, except that the expression levels of isoforms follow the geometric distribution with parameter $p = 0.8$.

Table 3.5 shows the effect of different distribution models on transcriptome assembly (in terms of sensitivity and precision) and isoform expression level estimation (in terms of R^2 value between estimated and real expression levels). In the log-normal model, only a few isoforms are assigned a high expression level, but in the geometric model, the expression levels of isoforms are more balanced. As a result, in the geometric model, it is more likely for a gene to have more than one highly expressed isoform. The sensitivity and precision of both Cufflinks and CEM are higher in the geometric model than in the log-normal model, but the R^2 values become much smaller. This shows that although both programs are able to reconstruct more isoforms correctly as more isoforms are highly expressed in a gene, they are unable to accurately estimate the expression levels of the isoforms.

Chapter 4

Accurate Isoform Inference and Abundance Estimation from Multiple Sample RNA-Seq Data

4.1 Introduction

In this chapter, we present a new algorithm for *ab initio* transcriptome assembly that is able to handle noisy RNA-Seq reads and multiple sample RNA-Seq datasets effectively. Instead of assembling transcripts separately for each sample and merging them together, our algorithm, called ISP (for *Iterative Shortest Path*), reconstructs transcripts directly from multiple samples. In fact, it takes advantage of the extra information contained in multiple sample RNA-Seq datasets (*e.g.*, correlation among the samples) to improve the performance of transcriptome assembly. By solving a linear programming problem iteratively in a weighted graph derived from the given multiple sample RNA-Seq datasets, ISP achieves a high performance by discarding problematic

reads and recovering missing junctions caused by various errors. Our preliminary experimental results on both simulated and real datasets and comparison with the popular assembly tools (Cufflinks and Cuffmerge) demonstrate that (i) ISP is able to assemble transcriptomes with a greatly increased precision while keeping the same level of sensitivity, especially when many samples are involved, and (ii) the assembly results of ISP help improve downstream differential analysis.

The rest of this chapter is organized as follows. The framework of the algorithm ISP is introduced in Section 4.2, including the graph construction, the linear programming problem, and the approach to incorporate paired-end read information and to recover missing junctions. The experimental results are presented in Section 4.3, while Section 4.4 concludes this chapter.

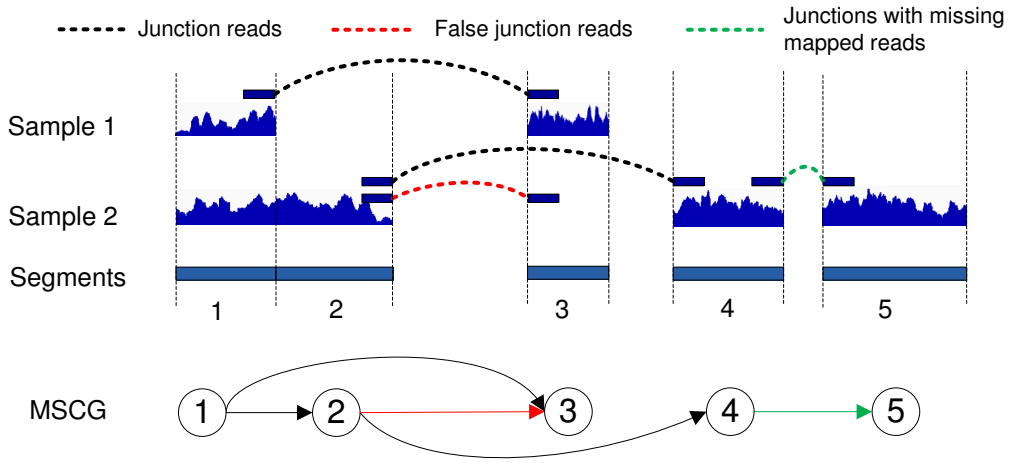


Figure 4.1: Transcriptome assembly from multiple sample RNA-Seq data.

4.2 Methods

4.2.1 Multiple Sample Connectivity Graph (MSCG)

A set of RNA-Seq reads from F different samples are first mapped independently to the reference genome using splice junction detection tools such as Tophat [86],

SpliceMap [2], *etc.* The mapped reads are then clustered into genes, and the exon-intron boundary information may be derived from either junction reads or existing annotations such as NCBI RefSeq [68] and UCSC known isoforms [36]. Based on this information, the sequence of a gene can be split into different *expressed segments* (or simply *segments*, [18]), where a segment is a continuous region in the reference genome uninterrupted by any splicing events (see Section 2.2.3 for a formal difinition of segment).

Several transcriptome assemblers use the *Connectivity Graph (CG)* to represent the splicing connections between segments or bases [26, 47] on single sample RNA-Seq data. Similarly, for multiple sample RNA-Seq data, we construct a *multiple sample connectivity graph (MSCG)* $G = (V, E)$ based on F sets of mapped RNA-Seq reads as follows. $V = \{v_i | 1 \leq i \leq M\}$ represents the M segments contained in a gene, and $(v_i, v_j) \in E$ if there is at least one read from the F datasets joining both segments i and j (see Figure 4.1).

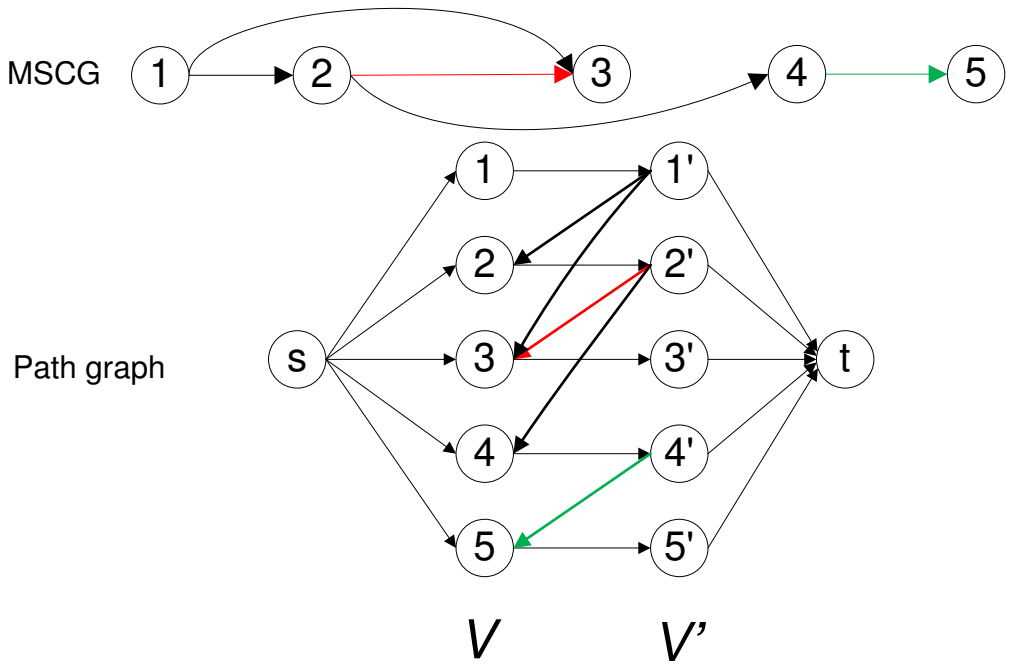


Figure 4.2: The construction of G_P from MSCG, using examples in Figure 4.1.

After constructing the MSCG graph, the average read density d_i for every vertex $v_i \in V$ and $d_{i,j}$ for every edge (v_i, v_j) is calculated as follows:

$$\begin{aligned} d_i &= \sum_{k=1}^{F} C_i^k/(l_v - L + 1) \\ d_{i,j} &= \sum_{k=1}^{F} C_{i,j}^k/(L-1) \end{aligned} \tag{4.1}$$

where C_i^k (or $C_{i,j}^k$) is the number of reads mapped to the corresponding segment (or junction, respectively) for the kth RNA-Seq dataset, l_v the length of the segment v and L the read length.

4.2.2 An Iterative Shortest Path (ISP) algorithm to find expressed isoforms

4.2.2.1 The path graph

Given an MSCG $G = (V, E)$ and the corresponding read density for each vertex and edge, we construct a *path graph* $G_P = (V_P, E_P)$ from G as follows. $V_P = V \cup V' \cup \{s, t\}$, where $V' = \{v' | \forall v \in V\}$. The edge set $E_P = E_s \cup E_t \cup E_V \cup E_E$ consists of four different types of edges.

- $(s, v) \in E_s$ for every $v \in V$. These are "source" edges;
- $(v', t) \in E_t$ for every $v' \in V'$. These are "sink" edges;
- $(v_i, v_i') \in E_V, i = 1, \cdots M$;
- For each edge $e = (v_i, v_j) \in E$, $(v_i', v_j) \in E_E$.

Figure 4.2 shows an example of construction G_p from G.

For simplicity, s is assigned number 0, t is assigned number $M+1$, and the vertices in V and V' are all assigned numbers 1 through M. Thus, an edge in E_P can be represented as (i,j), $0 \leq i,j \leq M+1$. For example, $(0,i) = (s,v_i) \in E_s$ and $(i,M+1) = (v'_i,t) \in E_t$ for $1 \leq i \leq M$. Similarly, $(i,i) \in E_V$ and $(i,j) \in E_E$ if $(v_i,v_j) \in E$, where $1 \leq i,j \leq M$.

4.2.2.2 Weighting the path graph

A weight $w_{i,j}$ is assigned for each edge $(i,j) \in E_P$ to reflect the likelihood that the corresponding segment (or junction) is problematic. A higher weight is assigned if the segment (or junction) is more likely due to noisy mapped reads. Notice that $w_{i,j}$ may be either positive (considered as "cost") or negative (considered as "reward").

For every edge $(0,i) \in E_s$, we "inactivate" the edge if v_i can be reached from another vertex v_j in the MSCG G:

$$w_{0,i} = \begin{cases} \infty & \text{if there exists } (j,i) \in E_E \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

Similarly, the edge $(i,M+1) \in E_t$ is disconnected if there is an edge from v_i to another vertex v_j in the MSCG G:

$$w_{i,M+1} = \begin{cases} \infty & \text{if there exists } (i,j) \in E_E \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

For every edge $(i,i) \in E_V$,

$$w_{i,i} = -\log(d_i + 1) \tag{4.4}$$

where d_i is the average read density of the segment corresponding to vertex v_i of the

MSCG G. Since we will look for a shortest path, paths going through segments with higher densities are preferred.

Since noisy junctions may result in incorrect assembly results, a higher positive cost is assigned for junction edges that are more likely to be problematic. For every edge $(i, j) \in E_E$, where $1 \leq i, j \leq M$ and $i \neq j$, we set

$$w_{i,j} = -\log P(i,j) = -\log(\frac{d_{i,j}}{\sum_k d_{i,k}} \frac{d_{i,j}}{\sum_k d_{k,j}}) \tag{4.5}$$

where $P(i, j)$ represents the probability of the junction between segments i and j being included in an isoform. Recall that $d_{i,j}$ is the average read density of edge $(v_i, v_j) \in G$.

4.2.2.3 The iterative shortest path problem

Given the path graph G_P, its corresponding edge weights and a set of single-end or paired-end reads R, we formulate the following linear programming (LP) problem, which is essentially a shortest path problem:

$$max \quad \sum_{0 \leq i,j \leq M+1} -w_{i,j} f(i,j) \tag{4.6}$$

$$\text{s.t.} \sum_{i=1}^{M} f(0,i) = 1, 1 \leq i \leq M \tag{4.7}$$

$$\sum_{0 \leq k \leq M} f(i,k) = \sum_{0 \leq k \leq M} f(k,i), 1 \leq i \leq M \tag{4.8}$$

$$0 \leq f(i,j) \leq 1, \quad 0 \leq i, j \leq M+1 \tag{4.9}$$

Equations (4.7)-(4.9) are constraints ensuring that the final solution represents a path (and thus an isoform) from s to t.

A gene may have multiple isoforms expressed, but only one isoform is ex-

tracted from solving the linear programming problem in Equation (4.17). To extract all expressed isoforms of a gene, we apply the "weight-decay" strategy [95] to modify the weights and iterate the algorithm several times. In each iteration, the weights are adjusted to encourage the algorithm to look for an isoform different from all previously found isoforms. The detailed algorithm is described in Algorithm 2.

Algorithm 2 ISP

Require: a path graph G_P and parameters W, the read set $R = \{r\}$, $\lambda > 1$, $\gamma > 1$.

1: Initialize the weights $w_{i,j}$, $0 \leq i, j \leq M + 1$; S $= \{\}$.
2: Solve the LP problem and find a path p from s to t with weight W_p.
3: **if** $W_p > W$ **then**
4: Terminate the algorithm and return S.
5: **end if**
6: Convert the path p to an isoform I, and remove all reads that are compatible with I in R.
7: **if** $I \notin$ S **then**
8: Set S $=$ S $\cup \{I\}$.
9: **end if**
10: **for** $0 \leq i, j \leq M + 1$ **do**
11: **if** $w_{i,j} \geq 0$ **then**
12: Set $w_{i,j} = \lambda w_{i,j}$;
13: **else**
14: $w_{i,j} = w_{i,j}/\lambda$.
15: **end if**
16: **end for**
17: **for** each edge $(i, j) \in E_E \cup E_V$ **do**
18: **if** the corresponding segment (or junction) is already used in S **then**
19: **if** $w_{i,j} \geq 0$ **then**
20: Set $w_{i,j} = \gamma w_{i,j}$;
21: **else**
22: Set $w_{i,j} = w_{i,j}/\gamma$.
23: **end if**
24: **end if**
25: **end for**
26: Go to Line 2.

The value of W determines how much error can be tolerated in the solution. A larger value of W tends to result in more isoforms (and thus recover more true ones), but isoforms with higher weights are more likely to be erroneous. The values of the parameters W, λ, γ can be determined empirically using simulation. In our experiments,

we set $W = 0$, $\lambda = 1.1$ and $\gamma = 1.1$.

4.2.3 Incorporating paired-end read information

Paired-end RNA-Seq reads provide more information than single-end reads since both pairs of the paired-end read come from the same fragment. To incorporate the paired-end read information into our framework, we modify the objective function (Equation (4.6)) to simultaneously minimize the cost of the path and maximize the number of paired-end reads that are *compatible* with the isoform implied by the path. Reads that are compatible with an isoform (introduced in Section 2.2.1.2) are the reads that are possibly generated from the isoform. If a read is compatible with an isoform, the splicing patterns implied by the read and the isoform must be identical. Specifically, a single-end read b containing k segments can be represented as a vector $b = (b_1, b_2, \cdots, b_k)$, where $1 \leq b_1 < \cdots < b_k \leq M$ are the segments included in b. An isoform I that b is compatible with must include all the segments $b_1, \cdots, b_k$, and must not include any other segment between b_1 and b_k. A paired-end read $p = (b, b')$ is compatible with I if and only if both b and b' are compatible with I.

For each paired-end read $p = (b, b')$, where $b = (b_1, \cdots b_k)$ and $b' = (b'_1, \cdots b'_k)$, we define the set of "inclusion segments" IS_p and "exclusion segments" ES_p as follows:

$$IS_p = b \cup b' \tag{4.10}$$

$$ES_p = \left\{ i : b_1 < i < b_k \text{ or } b'_1 < i < b'_k, i \notin IS_p \right\} \tag{4.11}$$

We modify the linear program in Section 4.2.2.3 to incorporate the information of paired-end reads as follows. For each paired-end read p, we define a binary variable

$q_p \in \{0, 1\}$ as whether p is compatible with the isoform implied in the solution. The objective function is modified as follows:

$$max \sum_{0 \leq i,j \leq M} -w_{i,j} f(i,j) + \alpha \sum_{p \in R} q_p \tag{4.12}$$

where $\alpha > 0$ is a user-defined parameter, and $f(i,j)$ and q_p satisfy the following constraints:

$$q_p, f(i,j) \in \{0,1\} \tag{4.13}$$

$$q_p = \prod_{i \in IS_p} f(i,i) \prod_{i \in ES_p} (1 - f(i,i)) \tag{4.14}$$

However, the integral and non-linear constraints make the problem difficult to solve. Instead, we relax the binary constraint for the variable $f(i,j)$ and q_p, and modify the constraint of q_p as follows:

$$0 \leq q_p, f(i,j) \leq 1, \tag{4.15}$$

$$q_p \leq f(i,i), i \in IS_p \tag{4.16}$$

$$q_p \leq 1 - f(i,i), i \in ES_p \tag{4.17}$$

By using the constraints in Equations (4.15)-(4.17), we can use linear program to solve the optimization problem defined in Equation (4.12). Equations (4.15)-(4.17) represent approximate constraints of Equations (4.13)-(4.14). If $f(i,i) \in \{0,1\}$, the upper bound of q_p is set to 1 by Equation (4.16) and (4.17) if the isoform implied in the solution is compatible with p. By maximizing Equation (4.12), the value of q_p is

therefore 1 if $\alpha > 0$. Otherwise if p is not compatible with the isoform, the upper bound will be set to 0 by Equation (4.16) or (4.17), and the only possible value of q_p is 0.

Ideally, the solution to the above problem is integral (*i.e.*, $f(i,j) \in \{0,1\}$), which represents a path (and an isoform) from s to t. However in some cases (about 0.1% of the genes in our simulated and real experiments), the linear program does not always guarantee an integral solution. For these genes, we solve the corresponding integer linear programming (ILP) problem by imposing an integral constraint to the variable $f(i,j)$ (*i.e.*, $f(i,j) \in \{0,1\}$). We use GNU Linear Programming Kit (GLPK, [105]) to solve the integer programming or integer linear programming problems.

4.2.4 Resolving ambiguities using Jensen-Shannon metric

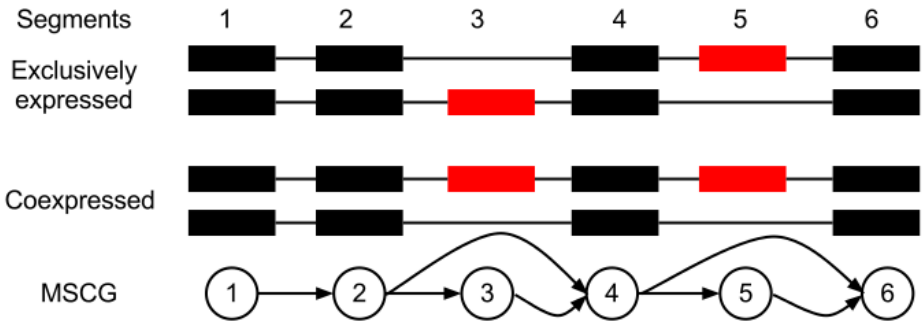


Figure 4.3: Coexpressed segments and exclusively expressed segments.

In complicated gene models, one MSCG may correspond to different sets of isoforms due to the segments that introduce ambiguities (named as "uncertain" segments). For example, the MSCG in Figure 4.3 includes two branches in segment 4, and different combinations of two uncertain segments, segments 3 and 5, introduce two possible sets of isoforms. Paired-end reads can be used to resolve the ambiguity (as in [26]), but it only works if there are paired-end reads mapped to uncertain segments.

In [87], isoforms are decomposed such that the expression levels of the segments in one isoform are similar, but this strategy does not consider the positional bias [99] and is applied only to a single sample.

In ISP, we use *Jensen-Shannon metric* (or JS metric) to resolve the ambiguity of uncertain segments. JS metric measures the similarity of the expression patterns between samples ([87]). If the JS metric of the expression levels of two uncertain segments is low (which means both segments are positively correlated), then both segments are likely to be included in one isoform (or they are "coexpressed segments", see Figure 4.3 as an example). Otherwise if the JS metric is high (meaning that both are negatively correlated), they are likely to appear in different isoforms (*i.e.*, they are "exclusively expressed segments").

JS metric is the square root of the *Jensen-Shannon divergence*, and is defined as follows:

$$JS(i,j) = \left(H(\frac{p_i + p_j}{2}) - \frac{H(p_i) + H(p_j)}{2} \right)^{1/2} \tag{4.18}$$

where $H(x)$ stands for the entropy of the probability distribution x. p_i is the distribution of segment i among samples, and is calculated based on the normalized number of reads in segment i among samples:

$$p_i = \left(\frac{p_i^1}{\sum_{k=1}^{F} p_i^k}, \cdots \frac{p_i^F}{\sum_{k=1}^{F} p_i^k} \right), \tag{4.19}$$

$$p_i^l = \frac{C_i^l}{C^l} \tag{4.20}$$

where C^l and C_i^l are the number of reads in the gene in sample l, and the number of reads in segment i in sample l, respectively (note that JS metric is used in [87] to measure the alternative splicing differences between samples).

To determine whether two uncertain segments are coexpressed or exclusively expressed, we first calculate two JS metric distributions from different sets of segments: P_{all}, the JS metrics between all pairs of segments in one gene, and P_{ref}, the JS metrics between pairs of some selected coexpressed segments. These coexpressed segments are selected by analyzing the MSCG of each gene: segments i and j are coexpressed if every path that includes i also includes j, and *vice versa*. After that, the JS metric of two uncertain segments i and j, $JS(i,j)$, is compared against both distributions. Segments i and j are considered exclusively expressed if $JS(i,j)$ is higher than the upper bound of the 95% confidence interval of P_{all}, and the following constraint is added to the linear programming (or integer programming) problem:

$$f(i,i) + f(j,j) \leq 1 \tag{4.21}$$

Otherwise if $JS(i,j)$ is smaller than $mean(P_{ref}) + std(P_{ref})$, segments i and j are considered coexpressed, and the following constraint is added:

$$f(i,i) - f(j,j) = 0 \tag{4.22}$$

4.3 Experimental results

4.3.1 Simulation results

We simulate RNA-Seq reads using the method described in Section 2.3.1 to evaluate the sensitivity and precision of ISP and Cufflinks on noisy RNA-Seq data and multiple samples. Isoforms from both UCSC human and mouse known genes are used as transcript models. UCSC human (mouse) known genes contain over $70,000$

(50,000) putative transcripts based on RefSeq, Genbank, CCDS, and UniProt, and is a moderately conservative set of transcripts [36]. We add two different types of noisy reads in the simulation to capture noises in real RNA-Seq data: *noisy junction reads* and *noisy intron reads*. Noisy junction reads are generated by randomly shifting the splicing positions of some normal junction reads by 1 to 3 bases. These reads are added since in reality, splicing regulators may shift the splice site a few bases to the proximal or distal intron boundaries [92, 54]. Noisy intron reads are reads coming randomly from the intron regions of a transcript. They are added since it has been observed that a fair amount of reads coming from intronic regions, possibly due to intron retention, non-coding RNAs or other unknown mechanisms [67].

4.3.1.1 The effect of noisy RNA-Seq reads on single sample data

We add different amounts of noisy reads of both types to a single sample RNA-Seq dataset. Figure 4.4 shows the sensitivity and precision of ISP and Cufflinks on various error rates using both human and mouse single-end reads, and Figure 4.5 shows the results of paired-end reads. Here, 80 million reads are used, and "x% error rate" means that x% of the junction reads are randomly shifted and x% of the intron reads are added. Both programs keep the same level of sensitivity (about 10%) while more erroneous reads are added, and Cufflinks achieves a higher precision than ISP when no erroneous reads are present. However, when more errors are added, the precision of both programs gradually drops, but ISP is less affected by the errors than Cufflinks. This comparison with respect to different error rates shows that ISP is able to handle read errors better than Cufflinks on single sample RNA-Seq data.

It is worth noting that when the simulated RNA-Seq data is error-free, mapping

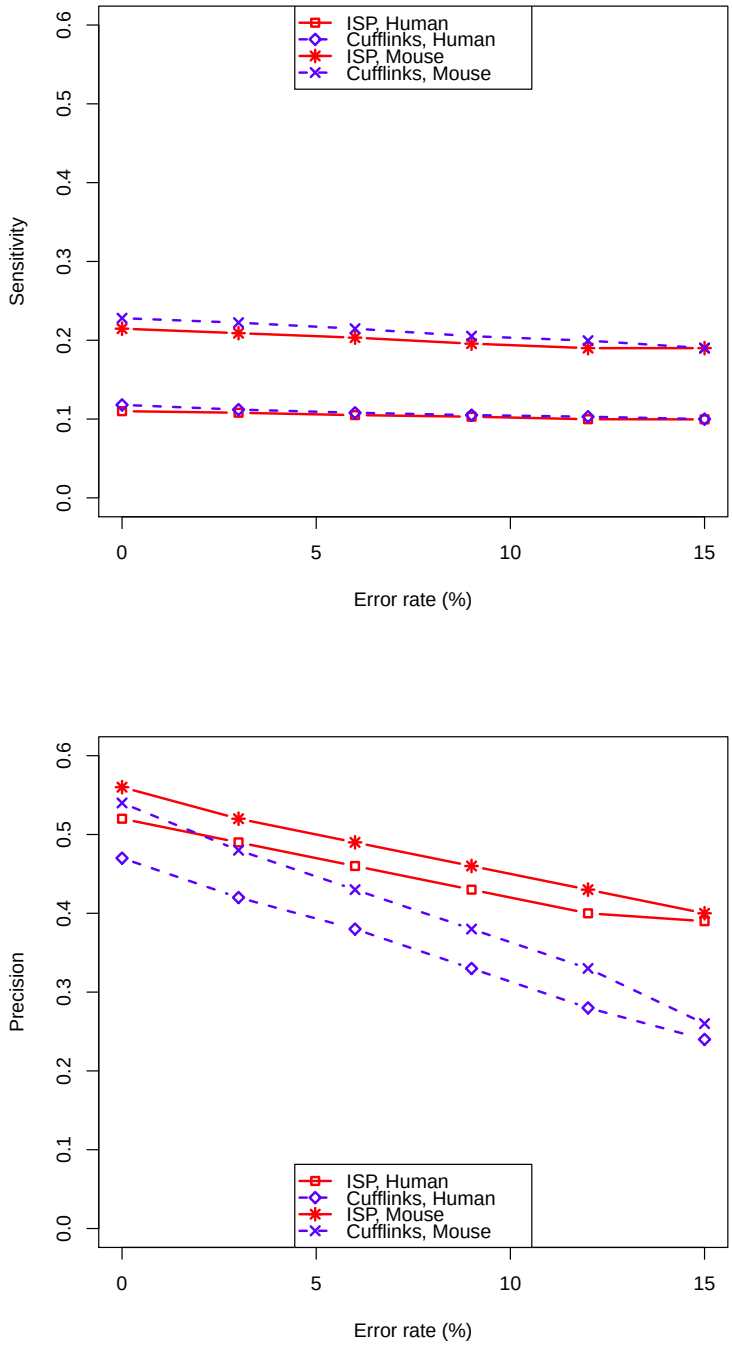


Figure 4.4: The sensitivity (top) and precision (bottom) of ISP and Cufflinks on a single RNA-Seq sample with various error rates. Errors come from either noisy junction reads or noisy intron reads.

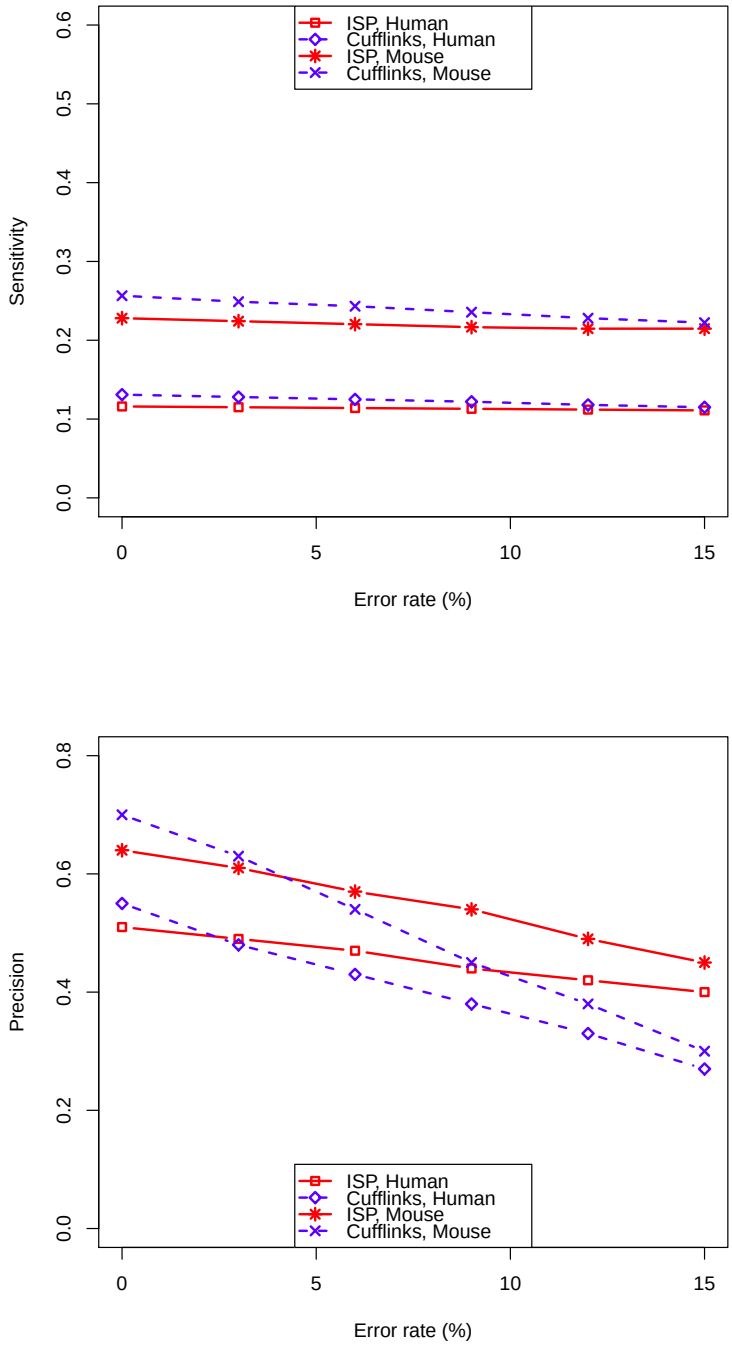


Figure 4.5: The sensitivity (top) and precision (bottom) of ISP and Cufflinks on a single RNA-Seq sample with various error rates using paired-end reads. Errors come from either noisy junction reads or noisy intron reads.

tools may still result in incorrectly mapped reads and thus the input to Cufflinks/ISP could still be noisy. Also, the low sensitivity of both programs is due to the fact that many of the transcripts are assigned very low expression level (or not expressed at all) based on the log-normal model [3]. These transcripts with few (or no) mapped reads decrease the value of sensitivity.

4.3.1.2 Assembly for multiple sample RNA-Seq data

To compare the performance of these algorithms on multiple sample RNA-Seq data, we generate six RNA-Seq datasets with different numbers of samples and evaluate the sensitivity and precision of the programs. For each dataset, the expression level of an isoform is independently assigned and 10% noisy reads are added as errors. To reconstruct all isoforms from multiple samples, a straightforward algorithm is to merge the RNA-Seq reads from all samples together and apply a transcriptome assembly tool (such as Cufflinks) designed for single sample RNA-Seq data. As a comparison to our ISP algorithm and Cuffmerge (which assembles transcripts for each sample separately and then merges them together), we also test Cufflinks and ISP on pooled data where RNA-Seq reads from all samples are merged together.

Figure 4.6 shows both sensitivity and precision of the four programs on different numbers of samples. When only one sample is considered, the sensitivity of all programs is the same. As more samples are added, all algorithms output more correct transcripts and improve their sensitivity, and both ISP and Cuffmerge achieve similar levels of sensitivity on six samples. As for the precision, ISP has a clear advantage, maintaining 40% to 60% higher values than Cuffmerge, and 60% to 80 % higher values than Cufflinks. The increasing trend of sensitivity and precision for both ISP and Cuffmerge shows

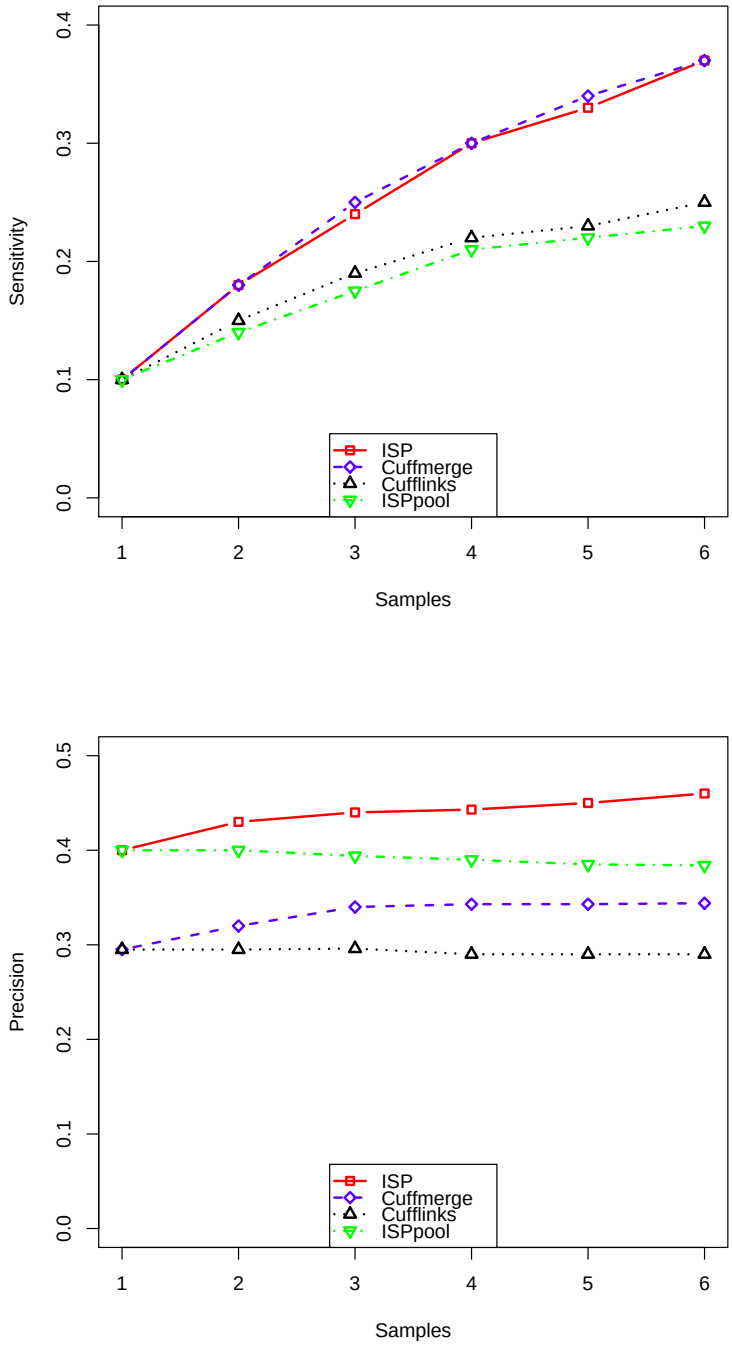


Figure 4.6: The sensitivity (top) and precision (bottom) of ISP and Cuffmerge on multiple samples, and on the pooled data (denoted as ISPpool and Cufflinks). The pooled data were generated by merging reads from all samples.

that both programs are able to take advantage of the existence of multiple samples and improve their sensitivity and precision simultaneously as more samples are used. Instead, the precisions of Cufflinks and ISP on the pooled data (Cufflinks and ISPpool) drop slightly while their rates of increase in sensitivity fall behind those of ISP and Cuffmerge. This is because as reads from more samples are merged, the splicing patterns become more complicated. Although more isoforms can be discovered (thus increasing its sensitivity), many incorrect isoforms are also predicted because of the increased difficulty to untangle the splicing patterns (thus slighly decreasing its precision). Therefore, the straightforward approach for dealing with multiple samples is not a good way to treat multiple sample RNA-Seq data.

4.3.1.3 Transcriptome assembly and differential analysis

In differential analysis, we are interested in finding and ranking genes (or isoforms) that are differentially expressed among two samples (or two groups of samples). Since isoforms assembled from individual samples may be different, it is necessary to construct a "universal" set of isoforms from all samples, from which the expression level estimation and statistical analysis can be performed. For example, Cufflinks includes a set of programs [87] for differential analysis, and all of them are based on merging isoforms from individual assemblies (using Cuffmerge).

We are interested in the effect of multiple sample transcriptome assembly on differential analysis. We simulate two RNA-Seq datasets and generate a set of isoforms for both samples by running (i) ISP and (ii) Cufflinks followed by Cuffmerge. To avoid using different expression level estimation methods preferred by both methods, we use Cuffdiff to estimate isoform expression levels and perform differential analysis (*i.e.*,

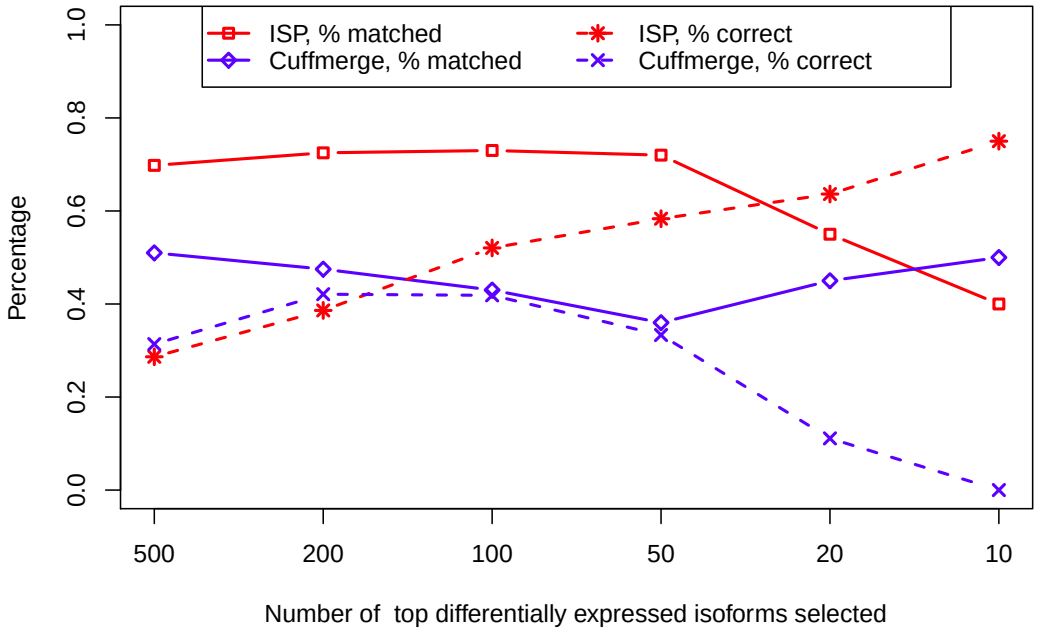


Figure 4.7: The performance of both ISP and Cuffmerge on differential analysis. Here, different numbers of top differentially expressed isoforms are considered. "% matched" is the percentage of these isoforms matched to UCSC human known isoforms and "% correct" is the percentage of the matched isoforms that have correct fold-change estimations (within range $[-2, +2]$ of the true fold change).

calculating the p value and q value).

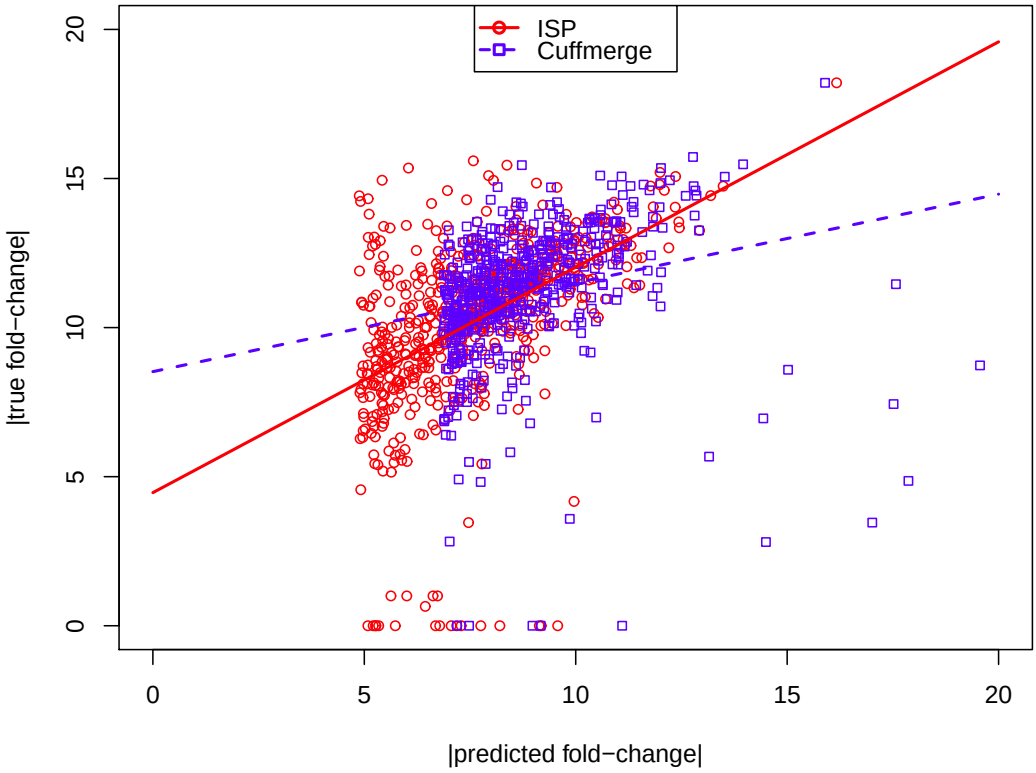


Figure 4.8: The true and predicted fold changes obtained by ISP and Cuffmerge. The two lines are the least squared regression lines for ISP ($y = 0.76x + 4.5$) and Cuffmerge ($y = 0.30x + 8.5$).

We rank the isoforms according to their predicted expression level fold changes between the two samples, and select different numbers of isoforms that show the greatest changes of expression levels. Figure 4.7 shows the percentage of isoforms that are matched to UCSC human known genes, and the percentage of matched ones that have correct fold-change estimations. Here, a correct fold-change estimation is defined as an estimated fold change within the [-2,+2] range of the true fold change. We show the trends as we decrease the number of selected isoforms, since we usually prefer finding fewer isoforms that are more differentially expressed between samples.

ISP found a larger number of matched (*i.e.*, true) isoforms than Cuffmerge when more than 10 isoforms are selected. This is expected since ISP has a higher precision than Cuffmerge, as shown in the previous experiments. Furthermore, ISP outputs more isoforms with correct fold-change estimations as fewer isoforms are selected. For the top 10 and 20 ranked isoforms, 80% of the isoforms predicted by ISP had their fold changes correctly estimated. On the other hand, the assembly results from Cuffmerge led to less accurate fold-change estimations, since the percentage of correct fold-change estimations decreased to a much lower level than ISP.

The better differential analysis results of ISP are further validated in Figure 4.8, where the estimated fold changes are compared to the true fold-change values. Here, the results of 1000 top isoforms that are differentially expressed and are matched to UCSC human known isoforms are shown. As can be seen from the figure, Cuffmerge predicts the fold changes of a few transcripts much higher than their true values. The regression lines between the predicted and true fold-change values for both programs also show that ISP is able to estimate the fold-change values closer to the truth.

Because we use the same algorithm (Cuffdiff) to estimate the expression levels of isoforms, we suspect that the low precision of Cuffmerge assembly led to the low accuracy in expression level estimation, hence reducing its performance in fold-change estimation.

4.3.2 Real RNA-Seq data results

To compare the performance of the algorithms on real RNA-Seq data, we use the public RNA-Seq datasets of 7 cancer cell lines downloaded from the ENCODE project [82]. These cell lines (GM12878, H1-hESC, K562, HeLa-S3, HepG2, HUVEC, NHEK; NCBI GEO accession code: GSE23316) include normal and cancer cells of differ-

ent tissues, and are the major cell models extensively used in biological and biomedical research [83].

4.3.2.1 Transcriptome assembly results

It is difficult to measure exactly which isoform is expressed in real RNA-Seq data since the current experimental techniques limit the ability to detect full-length transcripts efficiently. However, to evaluate the performance of transcript assembly algorithms on real RNA-Seq data, we argue that calculating both sensitivity and precision with respect to UCSC human known isoforms is a reasonable approach.

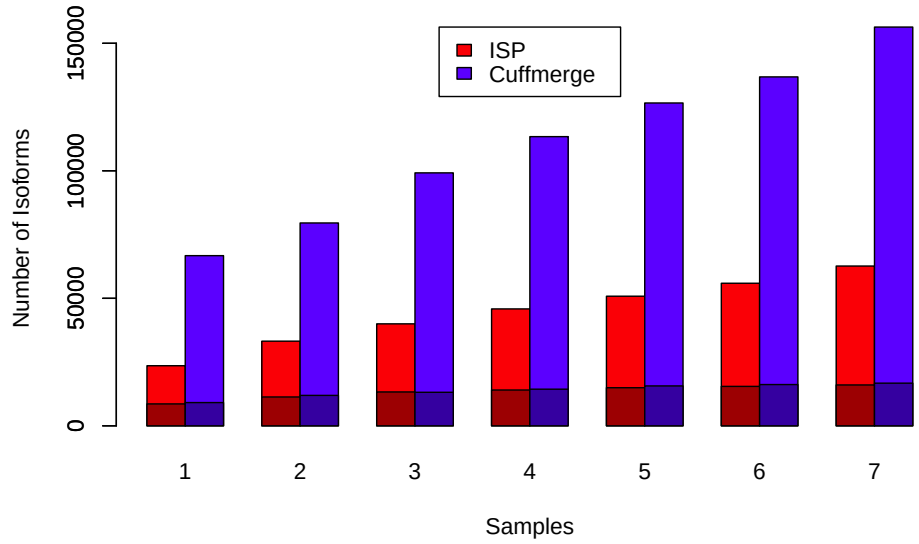


Figure 4.9: The number of predicted isoforms by ISP and Cuffmerge using multiple samples. The shaded region at the bottom of the bar shows the number of predicted isoforms that match UCSC human known isoforms.

Figure 4.9 shows the numbers of predicted isoforms together with the numbers of predictions matching UCSC human known isoforms for both programs by using different numbers of samples. For a single sample, the number of isoforms predicted by

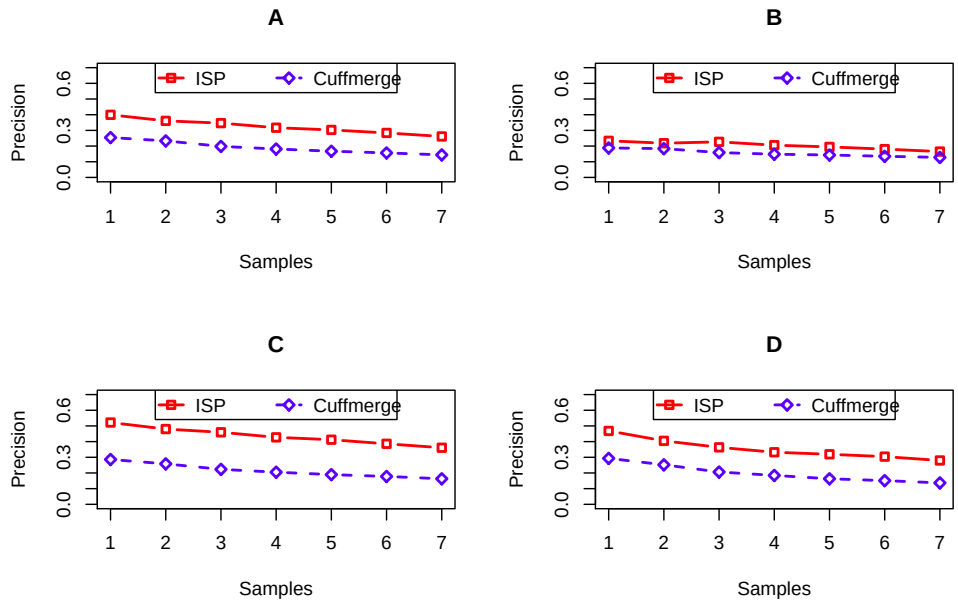


Figure 4.10: The precisions of both alorithms on multi-exon isoforms using all samples (A), and the precisions of both algorithms on isoforms grouped by the number of exons (B-D). B, C and D show the corresponding precisions for isoforms with 2-4, 5-10 and over 10 exons, respectively.

Cuffmerge's is over $60,000$, which is approximately twice the number of the isoforms predicted by ISP. As more RNA-Seq samples are added, more transcripts are merged by Cuffmerge, and this number reaches $150,000$ (over 100% growth) when all seven samples are included. In contrast, ISP shows a moderate increase in the number of predicted isoforms, with only 40% more isoforms predicted for seven samples compared to using only one sample. However, the number of isoforms that match UCSC human known isoforms remains roughly the same for both programs, with ISP accounting for over 90% of the numbers achieved by Cuffmerge. This illustrates that ISP is able to keep a high precision while sacrificing sensitivity a little when the number of samples increases.

Cuffmerge predictions include a large number of single-exon transcripts that do not match any UCSC human known isoforms. To study the effect of multiple samples on the inference of isoforms containing more than 1 exon, we exclude these single-exon

transcripts and calculate the precisions for isoforms grouped by their numbers of exons (Figure 4.10). The values of precision decrease as more samples (and thus reads) are added. This is different from our simulation results where a higher precision is obtained for all programs when using more samples. The reason might be that our simulation model is still too simple to capture the noise and errors in real RNA-Seq data.

ISP shows a higher precision than Cuffmerge on all multi-exon isoforms, and when all seven samples are used, the precision is doubled compared to Cuffmerge (Figure 4.10 A) for isoforms containing 5-10 exons. For isoforms with more than 10 exons, the difference between the two algorithms becomes smaller, but ISP still maintains a 70% higher precision than Cuffmerge(Figure 4.10 C and D). Isoforms with more exons are difficult to assemble since more errors may occur around splice junctions, and junction reads are more likely to be missing. As a result, the high precision of ISP may be attributed to its ability to handle noises effectively and to recover missing junctions by correlating multiple samples.

Some coexpressed (and exclusively expressed) uncertain segments are detected from 7 RNA-Seq samples, using the method described in Section 4.2.4. Figure 4.11 shows an example of the detected coexpressed segments in gene MEGF6. MEGF6 has two exon-skipping events, one skips exon A and the other skips exons B and C (marked in Figure 4.11). Both distributions of P_{all} and P_{ref} for MEGF6 are statistically different ($P < 0.002$, Wilcoxon rank-sum test). Exon A and exons B, C are determined coexpressed based on the JS metrics between exons A and B and between exons A and C. This is confirmed by the two UCSC known isoforms of MEGF6, one of which includes all the exons and the other includes none of them.

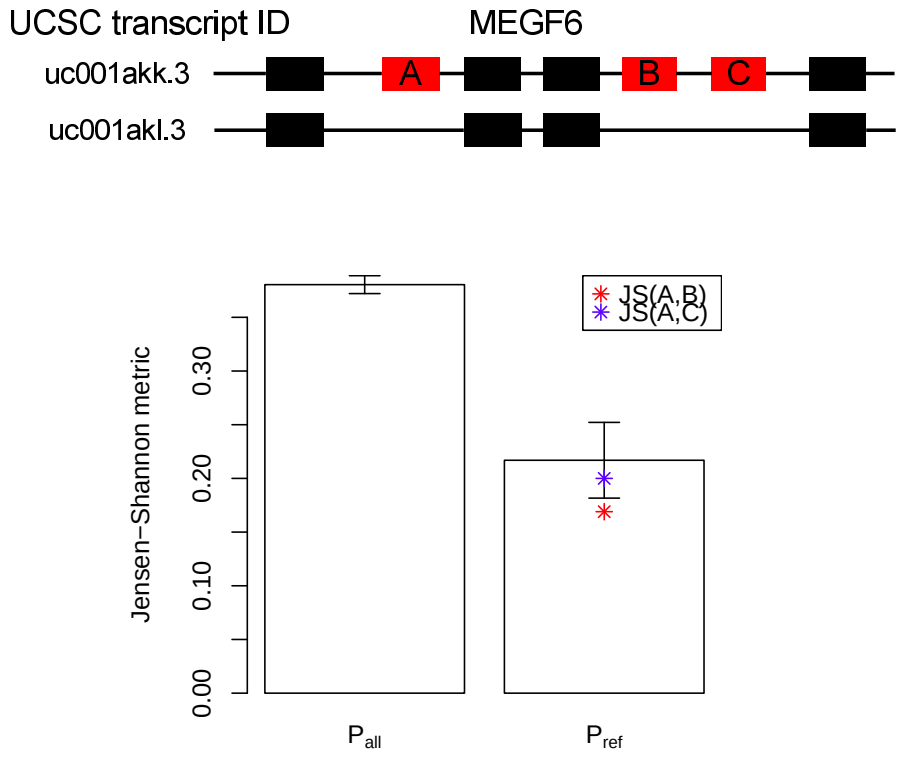


Figure 4.11: Part of the MEGF6 gene that includes two exon-skipping events (top), and the distributions of the Jensen-Shannon metrics (P_{all} and P_{ref}, bottom). P_{all} and P_{ref} are statistically different ($P < 0.002$, Wilcoxon rank-sum test), and the Jensen-Shannon metrics between exons A and B and between exons A and C are marked.

4.3.2.2 Differential analysis

The expression profiles of some ENCODE cell lines are measured by both RNA-Seq and Affymetrix Human Exon 1.0 ST Array (NCBI GEO accession code: GSE19090). We assemble transcripts from the RNA-Seq reads of two cell lines (GM12878 and K562) that have corresponding microarray experiments. Based on the assembly results, the expression levels of transcripts are calculated and compared with the microarray measurements.

An Affymetrix Human Exon Array uses "probesets" (*i.e.*, sets of probes) to measure the expression levels of exons. To calculate the expression levels of transcripts, we only keep probesets whose measured exons correspond to only one RefSeq transcript (called "unique" probesets). For those RefSeq transcripts that include at least one such unique probeset, their expression levels are calculated by averaging the measurements of all unique probesets. As in the simulated experiments, isoforms are reconstructed by ISP and Cuffmerge separately, and Cuffdiff is used for expression level estimation and differential analysis. Only isoforms that are matched to RefSeq transcripts are included in the comparison.

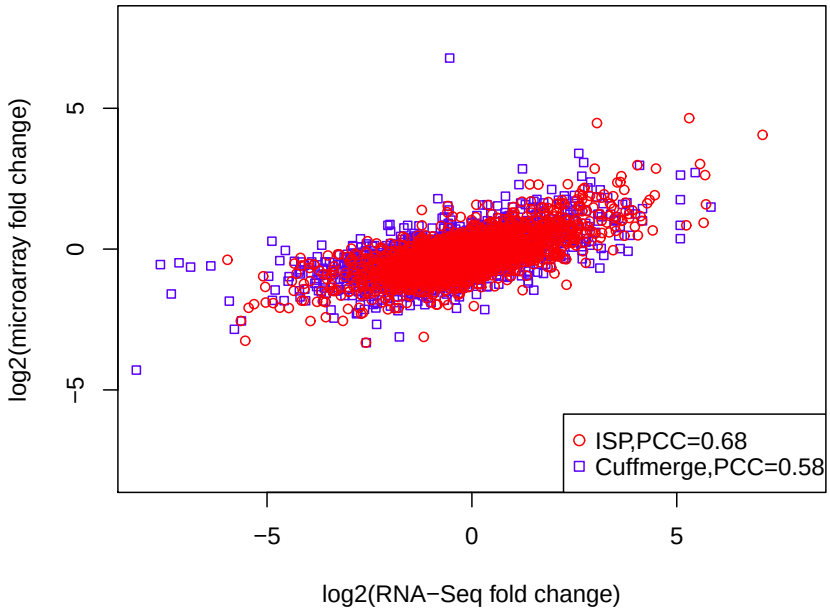


Figure 4.12: The estimation of transcript expression fold changes between two cell lines (GM12878 and K562) using both RNA-Seq and Affymetrix Human Exon 1.0 ST Array data.

Table 4.1: The correlation to microarray fold-change calculations, and the number of isoforms that are differentially expressed in microarray measurements among top ranked isoforms.

Top transcripts		10	50	100	200
PCC[1]	ISP	0.95	0.86	0.87	0.86
	Cuffmerge	0.89	0.82	0.82	0.81
confirmed ($p < 0.05$)[2]	ISP	9	45	94	170
	Cuffmerge	8	43	82	148
confirmed ($p < 0.001$)[2]	ISP	8	35	77	135
	Cuffmerge	7	32	64	108

[1]The Pearson Correlation Coefficient of the fold changes calculated by RNA-Seq and Affymetrix Human Exon 1.0 ST Array.
[2] The number of isoforms that are differentially expressed, confirmed by microarray data.

ISP and Cuffmerge detected 4468 and 4627 transcripts that have corresponding expression level estimations from microarray data, respectively, and Figure 4.12 shows the correlations of fold-change calculations between microarray and RNA-Seq data. The fold changes detected by RNA-Seq are larger than the fold changes calculated by microarray, which is consistent with previous findings [21]. The fold-change calculations based on the assembly results of ISP and Cuffmerge are quite accurate, while ISP reaches a higher PCC (Pearson Correlation Coefficient) value.

To further compare the differential analysis results, we rank transcripts according to their expression level changes, and select the transcripts that show the largest fold changes between samples (similar to the simulated experiments). For the corresponding microarray measurements of transcripts, we use student's t-test to check the hypothesis that these transcripts are differentially expressed between samples. Table 4.1 shows the PCC values of fold-change calculations between RNA-Seq and microarray measurements, and the number of differentially expressed transcripts confirmed by microarray data. The fold-change calculations based on ISP assembly results are more accurate since they have higher PCC values than Cufflinks, and a higher number of

predictions confirmed by microarray using different p value cutoffs. This shows that by using the transcripts reconstructed from ISP, we are able to get a more accurate differential analysis result than Cuffmerge.

4.4 Conclusion

With the advance of second generation sequencing technologies, it is now possible to reconstruct full-length transcripts, estimate their expression levels, and compare the structural and quantitative differences between samples. Transcriptome assembly can also benefit from the existence of multiple sample RNA-Seq data. However, inherent RNA-Seq errors have a negative impact on the performance of transcriptome assembly methods, which may in turn affect downstream differential analysis. In this chapter, we designed an algorithm (ISP) to reconstruct transcriptomes for multiple samples that is able to handle errors effectively.

RNA-Seq errors may come from various sources, for example chimeric RNA fragments during the RNA-Seq library preparation, and erroneous read mappings due to errors in both reads and the reference genome. These errors introduce a high false positive rate in the predicted isoforms. When combining assembly results from multiple samples, these false isoforms may accumulate and affect differential analysis. By using an iterative linear programming algorithm, ISP is able to discard erroneous segments or junctions, thus greatly improving the precision of the predictions. Furthermore, by using the multiple sample connectivity graph and by recovering missing junctions, our algorithm is able to make full use of the information among multiple samples to help assemble transcripts.

Both simulated and real experimental results show that, obtaining a set of ac-

curately assembled transcripts is crucial for downstream differential analysis. A large number of false positives decrease the accuracy of estimating the expression fold changes of isoforms between samples. ISP is able to achieve a better differential analysis performance by accurately assembling transcripts among samples.

Chapter 5

Conclusions and Future Work

RNA-Seq transcriptome assembly is an important and challenging problem in computational biology. In this dissertation, we discuss three different algorithms of RNA-Seq transcriptome assembly: IsoLasso, CEM and ISP. IsoLasso considers three different objectives (prediction accuracy, sparsity and completeness) simultaneously, and uses a modified LASSO regression algorithm to assemble transcripts and estimate transcript expression levels. CEM uses the quasi-multinomial distribution to handle different biases in RNA-Seq, and the negative Dirichlet distribution to reach sparsity. The probabilistic objective function is optimized by using a component elimination Expectation-Maximization (EM) algorithm. ISP builds a multiple sample connectivity graph (MSCG) directly from multiple sample RNA-Seq data, and solves a linear programming problem to accurately assemble transcripts from erroneous and multiple-sample RNA-Seq data. The performances of all three algorithms are validated by simulated and real RNA-Seq experiments.

Second generation sequencing (including RNA-Seq and RNA-Seq transcriptome assembly) is an active research area with many questions unanswered. For exam-

ple, almost all transcriptome assembly tools follow either the *ab initio* or the *de novo* approaches, and it remains unclear whether a new tool integrating both approaches will achieve a better performance. Although various mathematical models are proposed to correct RNA-Seq biases, the exact biological mechanism behind RNA-Seq biases is unknown. Further research efforts are required to obtain a more thorough understanding of RNA-Seq biases, and to develop new mathematical and statistical models to better describe RNA-Seq biases. Finally, new sequencing technologies, especially single molecule sequencing [17, 77, 20, 78], are posting even more challenges in computational biology, and new algorithms to address these challenges will be the key issue to bridge the gap between sequencing data and novel biological discoveries.

Bibliography

[1] Mark D. Alter, Daniel B. Rubin, Keri Ramsey, Rebecca Halpern, Dietrich A. Stephan, L. F. Abbott, and Rene Hen. Variation in the large-scale organization of gene expression levels in the hippocampus relates to stable epigenetic variability in behavior. *PLoS ONE*, 3(10):e3344, 10 2008.

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

[3] Martin Bengtsson, Anders Ståhlberg, Patrik Rorsman, and Mikael Kubista. Gene expression profiling in single cells from the pancreatic islets of Langerhans reveals lognormal distribution of mRNA levels. *Genome Research*, 15(10):1388–1392, October 2005.

[4] Manuele Bicego, Marco Cristani, and Vittorio Murino. Sparseness Achievement in Hidden Markov Models. In *Proceedings of the 14th International Conference on Image Analysis and Processing*, ICIAP '07, pages 67–72, Washington, DC, USA, 2007. IEEE Computer Society.

[5] David Brawand, Magali Soumillon, Anamaria Necsulea, Philippe Julien, Gábor Csárdi, Patrick Harrigan, Manuela Weier, Angélica Liechti, Ayinuer Aximu-Petri, Martin Kircher, Frank W. Albert, Ulrich Zeller, Philipp Khaitovich, Frank Grützner, Sven Bergmann, Rasmus Nielsen, Svante Pääbo, and Henrik Kaessmann. The evolution of gene expression levels in mammalian organs. *Nature*, 478(7369):343–348, October 2011.

[6] Ming-Sin Cheung, Thomas A. Down, Isabel Latorre, and Julie Ahringer. Systematic bias in high-throughput sequencing data and its correction by BEADS. *Nucleic Acids Research*, 39(15):e103, June 2011.

[7] Hamidreza Chitsaz, Joyclyn L. Yee-Greenbaum, Glenn Tesler, Mary-Jane Lombardo, Christopher L. Dupont, Jonathan H. Badger, Mark Novotny, Douglas B. Rusch, Louise J. Fraser, Niall A. Gormley, Ole Schulz-Trieglaff, Geoffrey P. Smith, Dirk J. Evers, Pavel A. Pevzner, and Roger S. Lasken. Efficient de novo assembly of single-cell bacterial genomes from short-read data sets. *Nature Biotechnology*, 29(10):915–921, September 2011.

[8] Nicole Cloonan, Alistair R R Forrest, Gabriel Kolle, Brooke B A Gardiner, Geoffrey J Faulkner, Mellissa K Brown, Darrin F Taylor, Anita L Steptoe, Shivangi

Wani, Graeme Bethel, Alan J Robertson, Andrew C Perkins, Stephen J Bruce, Clarence C Lee, Swati S Ranade, Heather E Peckham, Jonathan M Manning, Kevin J McKernan, and Sean M Grimmond. Stem cell transcriptome profiling via massive-scale mRNA sequencing. *Nat Meth*, 5(7):613–619, July 2008.

[9] P. C. Consul and G. C. Jain. A Generalization of the Poisson Distribution. *Technometrics*, 15(4):791–799, 1973.

[10] P. C. Consul and S. P. Mittal. A new urn model with predetermined strategy. *Biometrische Zeitschrift*, 17(2):67–75, 1975.

[11] P. C. Consul and S. P. Mittal. Some Discrete Multinomial Probability Models with Predetermined Strategy. *Biometrical Journal*, 19(3):161–173, 1977.

[12] Thomas H. Cormen, Charles E. Leiserson, and Ronald L. Rivest. *Introduction to Algorithms*, chapter 24.1, pages 651–655. MIT Press and McGraw-Hill, 2009.

[13] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[14] R. P. Dilworth. A decomposition theorem for partially ordered sets. *The Annals of Mathematics*, 51(1):pp. 161–166, 1950.

[15] Juliane C. Dohm, Claudio Lottaz, Tatiana Borodina, and Heinz Himmelbauer. Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Research*, 36(16):e105, September 2008.

[16] Bradley Efron, Trevor Hastie, Lain Johnstone, and Robert Tibshirani. Least angle regression. *Annals of Statistics*, 32:407–499, 2004.

[17] John Eid, Adrian Fehr, Jeremy Gray, Khai Luong, John Lyle, Geoff Otto, Paul Peluso, David Rank, Primo Baybayan, Brad Bettman, Arkadiusz Bibillo, Keith Bjornson, Bidhan Chaudhuri, Frederick Christians, Ronald Cicero, Sonya Clark, Ravindra Dalal, Alex deWinter, John Dixon, Mathieu Foquet, Alfred Gaertner, Paul Hardenbol, Cheryl Heiner, Kevin Hester, David Holden, Gregory Kearns, Xiangxu Kong, Ronald Kuse, Yves Lacroix, Steven Lin, Paul Lundquist, Congcong Ma, Patrick Marks, Mark Maxham, Devon Murphy, Insil Park, Thang Pham, Michael Phillips, Joy Roy, Robert Sebra, Gene Shen, Jon Sorenson, Austin Tomaney, Kevin Travers, Mark Trulson, John Vieceli, Jeffrey Wegener, Dawn Wu, Alicia Yang, Denis Zaccarin, Peter Zhao, Frank Zhong, Jonas Korlach, and Stephen Turner. Real-Time DNA Sequencing from Single Polymerase Molecules. *Science*, 323(5910):133–138, January 2009.

[18] Jianxing Feng, Wei Li, and Tao Jiang. Inference of isoforms from short sequence reads. In Bonnie Berger, editor, *Research in Computational Molecular Biology*, volume 6044 of *Lecture Notes in Computer Science*, pages 138–157. Springer Berlin / Heidelberg, 2010.

[19] M. A. F. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, March 2002.

[20] Benjamin A. Flusberg, Dale R. Webster, Jessica H. Lee, Kevin J. Travers, Eric C. Olivares, Tyson A. Clark, Jonas Korlach, and Stephen W. Turner. Direct detection of DNA methylation during single-molecule, real-time sequencing. *Nature methods*, 7(6):461–465, June 2010.

[21] Xing Fu, Ning Fu, Song Guo, Zheng Yan, Ying Xu, Hao Hu, Corinna Menzel, Wei Chen, Yixue Li, Rong Zeng, and Philipp Khaitovich. Estimating accuracy of RNA-Seq and microarrays with proteomics. *BMC Genomics*, 10(1):161+, April 2009.

[22] Pauline A. Fujita, Brooke Rhead, Ann S. Zweig, Angie S. Hinrichs, Donna Karolchik, Melissa S. Cline, Mary Goldman, Galt P. Barber, Hiram Clawson, Antonio Coelho, Mark Diekhans, Timothy R. Dreszer, Belinda M. Giardine, Rachel A. Harte, Jennifer Hillman-Jackson, Fan Hsu, Vanessa Kirkup, Robert M. Kuhn, Katrina Learned, Chin H. Li, Laurence R. Meyer, Andy Pohl, Brian J. Raney, Kate R. Rosenbloom, Kayla E. Smith, David Haussler, and W. James Kent. The UCSC Genome Browser database: update 2011. *Nucleic Acids Research*, 39(suppl 1):D876–D882, January 2011.

[23] Manfred G. Grabherr, Brian J. Haas, Moran Yassour, Joshua Z. Levin, Dawn A. Thompson, Ido Amit, Xian Adiconis, Lin Fan, Raktima Raychowdhury, Qiandong Zeng, Zehua Chen, Evan Mauceli, Nir Hacohen, Andreas Gnirke, Nicholas Rhind, Federica di Palma, Bruce W. Birren, Chad Nusbaum, Kerstin Lindblad-Toh, Nir Friedman, and Aviv Regev. Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nature Biotechnology*, 29(7):644–652, July 2011.

[24] Mika Gustafsson, Michael Hornquist, and Anna Lombardi. Constructing and analyzing a large-scale gene-to-gene regulatory network-lasso-constrained inference and biological validation. *IEEE/ACM Trans. Comput. Biol. Bioinformatics*, 2(3):254–261, 2005.

[25] Mitchell Guttman, Ido Amit, Manuel Garber, Courtney French, Michael F. Lin, David Feldser, Maite Huarte, Or Zuk, Bryce W. Carey, John P. Cassady, Moran N. Cabili, Rudolf Jaenisch, Tarjei S. Mikkelsen, Tyler Jacks, Nir Hacohen, Bradley E. Bernstein, Manolis Kellis, Aviv Regev, John L. Rinn, and Eric S. Lander. Chromatin signature reveals over a thousand highly conserved large non-coding RNAs in mammals. *Nature*, 458(7235):223–227, March 2009.

[26] Mitchell Guttman, Manuel Garber, Joshua Z. Levin, Julie Donaghey, James Robinson, Xian Adiconis, Lin Fan, Magdalena J. Koziol, Andreas Gnirke, Chad Nusbaum, John L. Rinn, Eric S. Lander, and Aviv Regev. Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincrnas. *Nature Biotechnology*, 28(5):503–510, May 2010.

[27] Brian J Haas and Michael C Zody. Advancing RNA-Seq analysis. *Nat Biotech*, 28(5):421–423, May 2010.

[28] Kasper D. Hansen, Steven E. Brenner, and Sandrine Dudoit. Biases in Illumina transcriptome sequencing caused by random hexamer priming. *Nucleic Acids Research*, 38(12):e131, July 2010.

[29] T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, chapter 3, page 57. Springer, 2009.

[30] R. R. Hocking and R. N. Leslie. Selection of the best subset in regression analysis. *Technometrics*, 9(4):531–540, 1967.

[31] Arthur E. Hoerl and Robert W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, 1970.

[32] Kathryn E. Holt, Julian Parkhill, Camila J. Mazzoni, Philippe Roumagnac, Francois-Xavier Weill, Ian Goodhead, Richard Rance, Stephen Baker, Duncan J. Maskell, John Wain, Christiane Dolecek, Mark Achtman, and Gordon Dougan. High-throughput sequencing provides insights into genome variation and evolution in salmonella typhi. *Nature Genetics*, 40(8):987–993, July 2008.

[33] G. C. Hon, R. D. Hawkins, O. L. Caballero, C. Lo, R. Lister, M. Pelizzola, A. Valsesia, Z. Ye, S. Kuan, L. E. Edsall, A. A. Camargo, B. J. Stevenson, J. R. Ecker, V. Bafna, R. L. Strausberg, A. J. Simpson, and B. Ren. Global DNA hypomethylation coupled to repressive chromatin domain formation and gene silencing in breast cancer. *Genome Research*, December 2011.

[34] Nobuaki Hoshino. On a limiting quasi-multinomial distribution. Technical Report CIRJE-F-361, CIRJE, Faculty of Economics, University of Tokyo, August 2005.

[35] Brian Howard and Steffen Heber. Towards reliable isoform quantification using RNA-SEQ data. *BMC Bioinformatics*, 11(Suppl 3):S6, 2010.

[36] Fan Hsu, W. James Kent, Hiram Clawson, Robert M. Kuhn, Mark Diekhans, and David Haussler. The ucsc known genes. *Bioinformatics*, 22(9):1036–1046, 2006.

[37] Aaron R. Jex, Shiping Liu, Bo Li, Neil D. Young, Ross S. Hall, Yingrui Li, Linfeng Yang, Na Zeng, Xun Xu, Zijun Xiong, Fangyuan Chen, Xuan Wu, Guojie Zhang, Xiaodong Fang, Yi Kang, Garry A. Anderson, Todd W. Harris, Bronwyn E. Campbell, Johnny Vlaminck, Tao Wang, Cinzia Cantacessi, Erich M. Schwarz, Shoba Ranganathan, Peter Geldhof, Peter Nejsum, Paul W. Sternberg, Huanming Yang, Jun Wang, Jian Wang, and Robin B. Gasser. Ascaris suum draft genome. *Nature*, 479(7374):529–533, October 2011.

[38] Hui Jiang and Wing H. Wong. Statistical inferences for isoform expression in rna-seq. *Bioinformatics*, 25(8):1026–1032, April 2009.

[39] Seyoung Kim, Kyung-Ah Sohn, and Eric P. Xing. A multivariate regression approach to association analysis of a quantitative trait network. *Bioinformatics*, 25(12):i204–i212, June 2009.

[40] Ben Langmead, Kasper Hansen, and Jeffrey Leek. Cloud-scale RNA-sequencing differential expression analysis with Myrna. *Genome Biology*, 11(8):R83, August 2010.

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

[42] Soohyun Lee, Chae H. Seo, Byungho Lim, Jin O. Yang, Jeongsu Oh, Minjin Kim, Sooncheol Lee, Byungwook Lee, Changwon Kang, and Sanghyuk Lee. Accurate quantification of transcriptome from RNA-Seq data by effective length normalization. *Nucleic Acids Research*, November 2010.

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

[44] Bo Li, Victor Ruotti, Ron M. Stewart, James A. Thomson, and Colin N. Dewey. RNA-Seq gene expression estimation with read mapping uncertainty. *Bioinformatics*, 26(4):493–500, February 2010.

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

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

[47] Wei Li, Jianxing Feng, and Tao Jiang. IsoLasso: A LASSO Regression Approach to RNA-Seq Based Transcriptome Assembly. In Vineet Bafna and S. Sahinalp, editors, *Research in Computational Molecular Biology*, volume 6577 of *Lecture Notes in Computer Science*, chapter 18, pages 168–188. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2011.

[48] Ryan Lister, Ronan C. O'Malley, Julian Tonti-Filippini, Brian D. Gregory, Charles C. Berry, A. Harvey Millar, and Joseph R. Ecker. Highly integrated Single-Base resolution maps of the epigenome in arabidopsis. *Cell*, 133(3):523 – 536, 2008.

[49] Shuangge Ma, Xiao Song, and Jian Huang. Supervised group lasso with applications to microarray data analysis. *BMC Bioinformatics*, 8(1):60+, February 2007.

[50] MAQC Consortium. The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nature biotechnology*, 24(9):1151–1161, September 2006.

[51] John C Marioni, Christopher E Mason, Shrikant M Mane, Matthew Stephens, and Yoav Gilad. RNA-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome Research*, 18(9):1509–1517, 2008.

[52] Jeffrey Martin, Vincent Bruno, Zhide Fang, Xiandong Meng, Matthew Blow, Tao Zhang, Gavin Sherlock, Michael Snyder, and Zhong Wang. Rnnotator: an automated de novo transcriptome assembly pipeline from stranded RNA-Seq reads. *BMC Genomics*, 11(1):663+, 2010.

[53] Jeffrey A. Martin and Zhong Wang. Next-generation transcriptome assembly. *Nature Reviews Genetics*, 12(10):671–682, September 2011.

[54] Arianne J. Matlin, Francis Clark, and Christopher W. Smith. Understanding alternative splicing: towards a cellular code. *Nature reviews. Molecular cell biology*, 6(5):386–398, May 2005.

[55] Roger Mclendon, Allan Friedman, Darrell Bigner, Erwin G. G. Van Meir, Daniel J. J. Brat, Gena M. Mastrogianakis, Jeffrey J. J. Olson, Tom Mikkelsen, Norman Lehman, Ken Aldape, Alfred W. K. K. Yung, Oliver Bogler, Scott Vandenberg,

Mitchel Berger, Michael Prados, Donna Muzny, Margaret Morgan, Steve Scherer, Aniko Sabo, Lynn Nazareth, Lora Lewis, Otis Hall, Yiming Zhu, Yanru Ren, Omar Alvi, Jiqiang Yao, Alicia Hawes, Shalini Jhangiani, Gerald Fowler, Anthony San Lucas, Christie Kovar, Andrew Cree, Huyen Dinh, Jireh Santibanez, Vandita Joshi, Manuel L. L. Gonzalez-Garay, Christopher A. A. Miller, Aleksandar Milosavljevic, Larry Donehower, David A. A. Wheeler, Richard A. A. Gibbs, Kristian Cibulskis, Carrie Sougnez, Tim Fennell, Scott Mahan, Jane Wilkinson, Liuda Ziaugra, Robert Onofrio, Toby Bloom, Rob Nicol, Kristin Ardlie, Jennifer Baldwin, Stacey Gabriel, Eric S. S. Lander, Li Ding, Robert S. S. Fulton, Michael D. D. Mclellan, John Wallis, David E. E. Larson, Xiaoqi Shi, Rachel Abbott, Lucinda Fulton, Ken Chen, Daniel C. C. Koboldt, Michael C. C. Wendl, Rick Meyer, Yuzhu Tang, Ling Lin, John R. R. Osborne, Brian H. H. Dunford-Shore, Tracie L. L. Miner, Kim Delehaunty, Chris Markovic, Gary Swift, William Courtney, Craig Pohl, Scott Abbott, Amy Hawkins, Shin Leong, Carrie Haipek, Heather Schmidt, Maddy Wiechert, Tammi Vickery, Sacha Scott, David J. J. Dooling, Asif Chinwalla, George M. M. Weinstock, Elaine R. R. Mardis, Richard K. K. Wilson, Gad Getz, Wendy Winckler, Roel G. W. G. Verhaak, Michael S. S. Lawrence, Michael O'Kelly, Jim Robinson, Gabriele Alexe, Rameen Beroukhim, Scott Carter, Derek Chiang, Josh Gould, Supriya Gupta, Josh Korn, Craig Mermel, Jill Mesirov, Stefano Monti, Huy Nguyen, Melissa Parkin, Michael Reich, Nicolas Stransky, Barbara A. A. Weir, Levi Garraway, Todd Golub, Matthew Meyerson, Lynda Chin, Alexei Protopopov, Jianhua Zhang, Ilana Perna, Sandy Aronson, Narayan Sathiamoorthy, Georgia Ren, Jun Yao, Ruprecht, Hyunsoo Kim, Sek W. Kong, Yonghong Xiao, Isaac S. S. Kohane, Jon Seidman, Peter J. J. Park, Raju Kucherlapati, Peter W. W. Laird, Leslie Cope, James G. G. Herman, Daniel J. J. Weisenberger, Fei Pan, David Van Den Berg, Leander Van Neste, Joo M. Yi, Kornel E. E. Schuebel, Stephen B. B. Baylin, Devin M. M. Absher, Jun Z. Z. Li, Audrey Southwick, Shannon Brady, Amita Aggarwal, Tisha Chung, Gavin Sherlock, James D. D. Brooks, Richard M. M. Myers, Paul T. T. Spellman, Elizabeth Purdom, Lakshmi R. R. Jakkula, Anna V. V. Lapuk, Henry Marr, Shannon Dorton, Yoon G. Choi, Ju Han, Amrita Ray, Victoria Wang, Steffen Durinck, Mark Robinson, Nicholas J. J. Wang, Karen Vranizan, Vivian Peng, Eric Van Name, Gerald V. V. Fontenay, John Ngai, John G. G. Conboy, Bahram Parvin, Heidi S. S. Feiler, Terence P. P. Speed, Joe W. W. Gray, Cameron Brennan, Nicholas D. D. Socci, Adam Olshen, Barry S. S. Taylor, Alex Lash, Nikolaus Schultz, Boris Reva, Yevgeniy Antipin, Alexey Stukalov, Benjamin Gross, Ethan Cerami, Wei Q. Wang, Li-Xuan X. Qin, Venkatraman E. E. Seshan, Liliana Villafania, Magali Cavatore, Laetitia Borsu, Agnes Viale, William Gerald, Chris Sander, Marc Ladanyi, Charles M. M. Perou, Neil D. Hayes, Michael D. D. Topal, Katherine A. A. Hoadley, Yuan Qi, Sai Balu, Yan Shi, Junyuan Wu, Robert Penny, Michael Bittner, Troy Shelton, Elizabeth Lenkiewicz, Scott Morris, Debbie Beasley, Sheri Sanders, Ari Kahn, Robert Sfeir, Jessica Chen, David Nassau, Larry Feng, Erin Hickey, Jinghui Zhang, John N. N. Weinstein, Anna Barker, Daniela S. S. Gerhard, Joseph Vockley, Carolyn Compton, Jim Vaught, Peter Fielding, Martin L. L. Ferguson, Carl Schaefer, Subhashree Madhavan, Kenneth H. H. Buetow, Francis Collins, Peter Good, Mark Guyer, Brad Ozenberger, Jane Peterson, and Elizabeth Thomson. Comprehensive genomic characterization defines human glioblastoma genes and core pathways. *Nature*, 455(7216):1061–1068, September 2008.

[56] Ryan Morin, Matthew Bainbridge, Anthony Fejes, Martin Hirst, Martin Krzywinski, Trevor Pugh, Helen McDonald, Richard Varhol, Steven Jones, and Marco Marra. Profiling the HeLa s3 transcriptome using randomly primed cDNA and massively parallel short-read sequencing. *BioTechniques*, 45(1):81–94, July 2008. PMID: 18611170.

[57] Olena Morozova, Martin Hirst, and Marco A Marra. Applications of new sequencing technologies for transcriptome analysis. *Annual Review of Genomics and Human Genetics*, 10(1):135–151, 2009. PMID: 19715439.

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

[59] Ugrappa Nagalakshmi, Zhong Wang, Karl Waern, Chong Shou, Debasish Raha, Mark Gerstein, and Michael Snyder. The transcriptional landscape of the yeast genome defined by RNA sequencing. *Science*, 320(5881):1344–1349, June 2008.

[60] Marius Nicolae, Serghei Mangul, Ion Mandoiu, and Alex Zelikovsky. Estimation of alternative splicing isoform frequencies from RNA-Seq data. *Algorithms for Molecular Biology*, 6(1):9+, 2011.

[61] Bogdan Paşaniuc, Noah Zaitlen, and Eran Halperin. Accurate Estimation of Expression Levels of Homologous Genes in RNA-seq Experiments. In Bonnie Berger, editor, *Research in Computational Molecular Biology*, volume 6044 of *Lecture Notes in Computer Science*, chapter 26, pages 397–409. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2010.

[62] Qun Pan, Ofer Shai, Christine Misquitta, Wen Zhang, Arneet L. Saltzman, Naveed Mohammad, Tomas Babak, Henry Siu, Timothy R. Hughes, Quaid D. Morris, Brendan J. Frey, and Benjamin J. Blencowe. Revealing global regulatory features of mammalian alternative splicing using a quantitative microarray platform. *Molecular cell*, 16(6):929–941, December 2004.

[63] Mee Y. Park and Trevor Hastie. L1-regularization path algorithm for generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 69(4):659–677, September 2007.

[64] Yu Peng, Henry Leung, S. Yiu, and Francis Chin. T-IDBA: A de novo Iterative de Bruijn Graph Assembler for Transcriptome. In Vineet Bafna and S. Sahinalp, editors, *Research in Computational Molecular Biology*, volume 6577 of *Lecture Notes in Computer Science*, chapter 31, pages 337–338. Springer Berlin / Heidelberg, Berlin, Heidelberg, 2011.

[65] Dorothee Pflueger, Stéphane Terry, Andrea Sboner, Lukas Habegger, Raquel Esgueva, Pei-Chun Lin, Maria A. Svensson, Naoki Kitabayashi, Benjamin J. Moss, Theresa Y. MacDonald, Xuhong Cao, Terrence Barrette, Ashutosh K. Tewari, Mark S. Chee, Arul M. Chinnaiyan, David S. Rickman, Francesca Demichelis, Mark B. Gerstein, and Mark A. Rubin. Discovery of non-ETS gene fusions in human prostate cancer using next-generation RNA sequencing. *Genome Research*, 21(1):56–67, January 2011.

[66] J. K. Pickrell, J. C. Marioni, A. A. Pai, J. F. Degner, B. E. Engelhardt, E. Nkadori, J. B. Veyrieras, M. Stephens, Y. Gilad, and J. K. Pritchard. Understanding mechanisms underlying human gene expression variation with RNA sequencing. *Nature*, 464(7289):768–772, March 2010.

[67] Chris P. Ponting and T. Grant Belgard. Transcribed dark matter: meaning or myth? *Human Molecular Genetics*, 19(R2):R162–R168, October 2010.

[68] Kim D. Pruitt, Tatiana Tatusova, Garth R. Brown, and Donna R. Maglott. NCBI Reference Sequences (RefSeq): current status, new features and genome annotation policy. *Nucleic Acids Research*, 40(D1):D130–D135, January 2012.

[69] Hugues Richard, Marcel H. Schulz, Marc Sultan, Asja Nürnberger, Sabine Schrinner, Daniela Balzereit, Emilie Dagand, Axel Rasche, Hans Lehrach, Martin Vingron, Stefan A. Haas, and Marie-Laure Yaspo. Prediction of alternative isoforms from exon expression levels in RNA-Seq experiments. *Nucleic Acids Research*, 38(10):e112, June 2010.

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

[71] Mark Robinson and Alicia Oshlack. A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biology*, 11(3):R25+, March 2010.

[72] Joel Rozowsky, Ghia Euskirchen, Raymond K. Auerbach, Zhengdong D. Zhang, Theodore Gibson, Robert Bjornson, Nicholas Carriero, Michael Snyder, and Mark B. Gerstein. PeakSeq enables systematic scoring of ChIP-seq experiments relative to controls. *Nature Biotechnology*, 27(1):66–75, January 2009.

[73] Julia Salzman, Hui Jiang, and Wing H. Wong. Statistical Modeling of RNA-Seq Data. *Statistical Science*, 26(1):62–83, February 2011.

[74] M Sammeth, V Lacroix, P Ribeca, and R Guigo. The flux simulator. *http://flux.sammeth.net*, 2010.

[75] Schraga Schwartz, Ram Oren, and Gil Ast. Detection and Removal of Biases in the Analysis of Next-Generation Sequencing Reads. *PLoS ONE*, 6(1):e16685, January 2011.

[76] Jared T. Simpson, Kim Wong, Shaun D. Jackman, Jacqueline E. Schein, Steven J. Jones, and Inanç Birol. ABySS: a parallel assembler for short read sequence data. *Genome research*, 19(6):1117–1123, June 2009.

[77] Catherine C. Smith, Qi Wang, Chen-Shan Chin, Sara Salerno, Lauren E. Damon, Mark J. Levis, Alexander E. Perl, Kevin J. Travers, Susana Wang, Jeremy P. Hunt, Patrick P. Zarrinkar, Eric E. Schadt, Andrew Kasarskis, John Kuriyan, and Neil P. Shah. Validation of ITD mutations in FLT3 as a therapeutic target in human acute myeloid leukaemia. *Nature*, 485(7397):260–263, May 2012.

[78] Chun-Xiao X. Song, Tyson A. Clark, Xing-Yu Y. Lu, Andrey Kislyuk, Qing Dai, Stephen W. Turner, Chuan He, and Jonas Korlach. Sensitive and specific single-molecule sequencing of 5-hydroxymethylcytosine. *Nature methods*, 9(1):75–77, November 2011.

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

[80] The Cancer Genome Atlas Research Network. Integrated genomic analyses of ovarian carcinoma. *Nature*, 474(7353):609–615, June 2011.

[81] The ENCODE Project Consortium. The ENCODE (ENCyclopedia Of DNA Elements) Project. *Science*, 306(5696):636–640, October 2004.

[82] The ENCODE Project Consortium. Identification and analysis of functional elements in 1% of the human genome by the ENCODE pilot project. *Nature*, 447(7146):799–816, June 2007.

[83] The ENCODE Project Consortium. A User's Guide to the Encyclopedia of DNA Elements (ENCODE). *PLoS Biol*, 9(4):e1001046+, April 2011.

[84] The modENCODE Consortium. Identification of Functional Elements and Regulatory Circuits by Drosophila modENCODE. *Science*, 330(6012):1787–1797, December 2010.

[85] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

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

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

[88] P. Kerr Wall, Jim Leebens-Mack, Andre Chanderbali, Abdelali Barakat, Erik Wolcott, Haiying Liang, Lena Landherr, Lynn Tomsho, Yi Hu, John Carlson, Hong Ma, Stephan Schuster, Douglas Soltis, Pamela Soltis, Naomi Altman, and Claude dePamphilis. Comparison of next generation sequencing technologies for transcriptome characterization. *BMC Genomics*, 10(1):347, 2009.

[89] Lin Wan, Xiting Yan, Ting Chen, and Fengzhu Sun. Modeling RNA degradation for RNA-Seq with applications. *Biostatistics*, February 2012.

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

[91] Likun Wang, Zhixing Feng, Xi Wang, Xiaowo Wang, and Xuegong Zhang. DEGseq: an R package for identifying differentially expressed genes from RNA-seq data. *Bioinformatics*, 26(1):136–138, January 2010.

[92] Zefeng Wang, Xinshu Xiao, Eric Van Nostrand, and Christopher B. Burge. General and specific functions of exonic splicing silencers in splicing control. *Molecular cell*, 23(1):61–70, July 2006.

[93] Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature reviews. Genetics*, 10(1):57–63, January 2009.

[94] Zhong Wang, Mark Gerstein, and Michael Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nat Rev Genet*, 10(1):57–63, January 2009.

[95] P. J. Werbos. Backpropagation: past and future. In *Neural Networks, 1988., IEEE International Conference on*, pages 343–353 vol.1. IEEE, July 1988.

[96] David A. Wheeler, Maithreyan Srinivasan, Michael Egholm, Yufeng Shen, Lei Chen, Amy McGuire, Wen He, Yi-Ju Chen, Vinod Makhijani, G. Thomas Roth, Xavier Gomes, Karrie Tartaro, Faheem Niazi, Cynthia L. Turcotte, Gerard P. Irzyk, James R. Lupski, Craig Chinault, Xing-zhi Song, Yue Liu, Ye Yuan, Lynne Nazareth, Xiang Qin, Donna M. Muzny, Marcel Margulies, George M. Weinstock, Richard A. Gibbs, and Jonathan M. Rothberg. The complete genome of an individual by massively parallel dna sequencing. *Nature*, 452(7189):872–876, April 2008.

[97] Brian T. Wilhelm, Samuel Marguerat, Stephen Watt, Falk Schubert, Valerie Wood, Ian Goodhead, Christopher J. Penkett, Jane Rogers, and Jurg Bahler. Dynamic repertoire of a eukaryotic transcriptome surveyed at single-nucleotide resolution. *Nature*, 453(7199):1239–1243, June 2008.

[98] Tong T. Wu, Yi F. Chen, Trevor Hastie, Eric Sobel, and Kenneth Lange. Genome-wide association analysis by lasso penalized logistic regression. *Bioinformatics*, 25(6):714–721, March 2009.

[99] Zhengpeng Wu, Xi Wang, and Xuegong Zhang. Using non-uniform read distribution models to improve isoform expression inference in RNA-Seq. *Bioinformatics*, 27(4):502–508, February 2011.

[100] Xinshu Xiao, Zefeng Wang, Minyoung Jang, Razvan Nutiu, Eric T. Wang, and Christopher B. Burge. Splice site strength-dependent activity and genetic buffering by poly-G runs. *Nature structural & molecular biology*, 16(10):1094–1100, October 2009.

[101] Moran Yassour, Tommy Kaplan, Hunter B. Fraser, Joshua Z. Levin, Jenna Pfiffner, Xian Adiconis, Gary Schroth, Shujun Luo, Irina Khrebtukova, Andreas Gnirke, Chad Nusbaum, Dawn-Anne A. Thompson, Nir Friedman, and Aviv Regev. Ab initio construction of a eukaryotic transcriptome by massively parallel mrna sequencing. *Proceedings of the National Academy of Sciences of the United States of America*, 106(9):3264–3269, March 2009.

[102] Tjalling J. Ypma. Historical Development of the Newton-Raphson Method. *SIAM Review*, 37(4):531–551, 1995.

[103] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society Series B*, 67(2):301–320, April 2005.

[104] *Optimization Toolbox User's Guide*. The Mathworks, Inc., Natrik, MA, USA, 2004.

[105] GNU Linear Programming Kit (GLPK). http://www.gnu.org/software/glpk, 2008.