

RESEARCH

Fast and interpretable alternative splicing and differential gene-level expression analysis using transcriptome segmentation with Yanagi

Mohamed K Gunady^{1,2*}, Stephen M Mount^{2,3} and Héctor Corrada Bravo^{1,2}

*Correspondence:

mgunady@cs.umd.edu

¹Department of Computer Science, University of Maryland, College Park, Maryland, USA

²Center for Bioinformatics and Computational Biology, University of Maryland, College Park, Maryland, USA

Full list of author information is available at the end of the article

[†]Equal contributor

Abstract

Introduction: Analysis of differential alternative splicing from RNA-seq data is complicated by the fact that many RNA-seq reads map to multiple transcripts, besides, the annotated transcripts are often a small subset of the possible transcripts of a gene. Here we describe Yanagi, a tool for segmenting transcriptome to create a library of maximal L-disjoint segments from a complete transcriptome annotation. That segment library preserves all transcriptome substrings of length L and transcripts structural relationships while eliminating unnecessary sequence duplications.

Contributions: In this paper, we formalize the concept of transcriptome segmentation and propose an efficient algorithm for generating segment libraries based on a length parameter dependent on specific RNA-Seq library construction. The resulting segment sequences can be used with pseudo-alignment tools to quantify expression at the segment level. We characterize the segment libraries for the reference transcriptomes of *Drosophila melanogaster* and *Homo sapiens* and provide gene-level visualization of the segments for better interpretability. Then we demonstrate the use of segments-level quantification into gene expression and alternative splicing analysis. The notion of transcript segmentation as introduced here and implemented in Yanagi opens the door for the application of lightweight, ultra-fast pseudo-alignment algorithms in a wide variety of RNA-seq analyses.

Conclusion: Using segment library rather than the standard transcriptome succeeds in significantly reducing ambiguous alignments where reads are multimapped to several sequences in the reference. That allowed avoiding the quantification step required by standard kmer-based pipelines for gene expression analysis. Moreover, using segment counts as statistics for alternative splicing analysis enables achieving comparable performance to counting-based approaches (e.g. rMATS) while rather using fast and lightweight pseudo alignment.

Keywords: Transcriptome Quantification; Differential Gene Expression; Alternative Splicing; RNA-Seq; pseudo alignment; Segmentation

1 Introduction

Messenger RNA transcript abundance estimation from RNA-Seq data is a crucial task in high-throughput studies that seek to describe the effect of genetic or environmental changes on gene expression. Transcript-level analysis and abundance estimation can play a central role in performing fine-grained analysis studying local splicing events and coarse-grained analysis studying changes in gene expressions.

Over the years, various approaches have addressed the joint problems of (gene level) transcript expression quantification and differential alternative RNA processing. Much effort in the area has been dedicated to the problem of efficient alignment, or pseudo-alignment, of reads to a genome or a transcriptome, since this is typically a bottleneck in the analytical processes that start with RNA-Seq reads and yield gene-level expression or differentially expressed transcripts. Among these approaches are alignment techniques such as bowtie [1], Tophat [2, 3], and Cufflinks [4], and newer techniques such as sailfish [5], RapMap [6], Kallisto [7] and Salmon [8], which provide efficient strategies through k-mer counting that are much faster, but maintain comparable, or superior, accuracy.

These methods simplified the expected outcome of the alignment step to find only sufficient read-alignment information required by the quantification step. Given a transcriptome reference, an index of kmers is created and used to find a mapping between reads and the list of compatible transcripts based on each approach's definition of compatibility. The next step, quantification, would be to resolve the ambiguity in reads that were mapped to multiple transcripts. Multi-mapping reads are common even assuming error free reads, due to shared regions produced by alternative splicing. The ambiguity in mapping reads is resolved using probabilistic models, such as the EM algorithm, to produce the abundance estimate of each transcript [9]. It is at this step that transcript-level abundance estimation still face substantial challenges that inherently affect the underlying analysis.

The presence of sequence repeats and paralogous genes in many organisms creates ambiguity in the placement of reads. More importantly, the fact that alternatively spliced isoforms share substantial portions of their coding regions, greatly increases the proportion of reads coming from these shared regions and consequently reads being multi-mapped becomes more frequent when aligning to annotated transcripts (Figure 1 A-B). In fact, local splicing variations can be joined combinatorially to create a very large number of possible transcripts from many genes. An extreme case is the *Drosophila* gene *Dscam*, which can produce over 38,000 transcripts by joining less than 50 exons [10]. More generally, long-read sequencing indicates that although there are correlations between distant splicing choices [11], a large number of possible combinations is typical. Thus, standard annotations, which enumerate only a minimal subset of transcripts from a gene (e.g. [12]) are inadequate descriptions. Furthermore, short read sequencing, which is likely to remain the norm for some time, does not provide information for long-range correlations between splicing events.

In this paper, we propose a novel strategy that aims at constructing a set of transcriptome segments that can be used in the read-alignment-quantification steps instead of the whole transcriptome without loss of information. Such a set of segments (a segment library) can fully describe individual events (primarily local splicing variation, but also editing sites or sequence variants) independently, leaving the estimation of transcript abundances as a separate problem. Here we introduce and formalize the idea of transcriptome segmentation, propose and analyze an algorithm for transcriptome segmentation, through a tool called Yanagi. To show how the segments library can be used in downstream analysis, we show results of using yanagi for gene-level and alternative splicing differential analysis.

2 Transcriptome Segmentation

Figure 1 shows a typical situation in RNASeq data analysis and provides an overview of the transcript segmentation strategy. In particular, it summarizes how reads that would be multi-mapped when aligning to a transcript library would be aligned to segments. In the latter case, all reads are aligned to a single target sequence and read counts are obtained per segment without the need of probabilistic quantification methods to resolve ambiguity. The next few subsections present a few more specifics of the Yanagi [13] method for transcriptome segmentation.

2.1 Segments Properties

Yanagi's objective is to generate a minimal set of disjoint sequences (where disjointness is parameterized by the experimental sequencing read length) while maintaining transcriptome sequence completeness.

The following definitions are for a given transcriptome T , and parameter L .

Definition 1 A Segment

A segment seg defined by the tuple $\langle exs, loc, w \rangle$ is a genomic region of width w beginning at genomic location loc and spanning the sequence of consecutive exonic regions $exs \in Exs_T$ (either exons or retained introns). Exonic regions are considered consecutive if they are consecutively spliced into at least one possible isoform in T . And for all segments in a segments library $S_{T,L}$, its width w is at least L bases.

Definition 2 Segments Sequences Completeness

The set of segments $S_{T,L}$ is Complete if and only if

$$seq \in S_{T,L}; \forall seq \in Substring(T), len(seq) \leq L$$

and

$$seq \in Substring(T); \forall seq \in Substring(S_{T,L})$$

Definition 3 L-disjoint Segments

Each segment in the set $S_{T,L}$ is L-disjoint if and only if

$$width[overlap(seg_i, seg_j)] < L; \forall seg_i, seg_j \in S, i \neq j$$

The L-disjointness property restricts any pair of *L-disjoint* segments to have an overlap region shorter than parameter L , which typically equals to the sequencing read length. In other words no read of length at least L can be mapped to both segments of an *L-disjoint* segment pair, assuming error-free reads.

Another property of the generated segments is to be maximal. For $seg : \langle exs, loc, w \rangle$, denote $Txs(seg)$ as the set intersection of annotated transcripts splicing exons exs . We can define a subsumption relationship between segments as $seg_1 \succ seg_2$ if and only if $exs_1 = exs_2, loc_1 = loc_2, Txs(seg_1) = Txs(seg_2)$ and $w_1 > w_2$. With this relationship we can define the following property of a segment library $S_{T,L}$

Definition 4 *Maximal Segments*

For each segment in the set $S_{T,L}$ to be Maximal

$$seg_1 \succ seg_2 \Rightarrow seg_2 \notin S_{T,L}, \forall seg_1 \in S_{T,L}$$

Thus a maximal segment is the longest common sequence of genomic regions starting at *loc*, such that these regions are spliced similarly, i.e. the entire sequence belongs to the same set of transcripts. That is why in figure 1 (C) segment S5 is extended to include two exons and its junction, while segment S2 is interrupted by the different splicings of Tx1 and Tx2.

2.2 Segmentation Algorithm

The transcriptome segmentation process can be summarized into three steps: (1) Preprocessing the transcriptome annotation in order to obtain disjoint exonic bins, (2) Constructing a Segments Graph, and finally (3) Generating the final segments. Transactions in Figure 1 (F) represent these three steps.

1. *Annotation Preprocessing:*

Yanagi applies a preprocessing step to eliminate region overlaps present in the transcriptome reference. Parts of an exon (or a retained intron) can be differentially spliced between isoforms either due alternative 3'/5' splice sites, or transcription start/end sites. For example, splicing the first and second exons between Tx1 and Tx3 in figure 1 (F). This step ensures that any splicing event is occurring either at the beginning or the end of an exonic bin, which makes the process of generating maximal L-disjoint segments easier. The preprocessing step is independent from the parameter L , so it can be done only once per transcriptome reference.

2. *Constructing Segments Graph:*

Currently Yanagi builds a separate segment graph for each gene, since there are no alternative splicing events between transcripts of different genes. However, future work may use segment graphs that connect different genes sharing regions of identical sequence length L or greater, but we have yet to address this.

Definition 5 *Segments Graph*

A segment graph $G_{T,L}$ is an acyclic directed graph defined by the pair (N, E) , where N is a set of nodes representing segments, and E is the set of directed edges between the nodes. An edge $e : (n_i, n_j) \in E$ is created if the segment corresponding to node n_i directly precedes the segment corresponding to node n_j in some transcript.

For each gene, the preprocessed Splice graph is parsed to construct a set of segment nodes (review algorithm details in [13]). These nodes formulate the segments graph of that gene. Each segment node represents an L-disjoint segment, which is not necessarily a maximal segment.

3. *Generating Segments:*

To preserve the maximality property, the segments graph is parsed to aggregated segment nodes into the final maximal segments. In a segment graph, if there is an edge from $node_i$ to $node_j$ while $outdegree(node_i) = indegree(node_j) = 1$, that implies that both nodes belong to the same set of transcripts and can be aggregated

into a segment that subsumes both nodes. In other words, aggregating nodes along a path in the segment graph bounded by branching points (nodes with indegree or outdegree greater than 1).

Yanagi reports the segments into a FASTA file. Each sequence represents a maximal L-disjoint segment. Each segment sequence has a header specifying metadata of how each segment was formed, including: gene ID, the set of exonic bins *exs* included in the segment, genome location in the first exonic bin of *exs* where the segment starts, genome location in the last exonic bin of *exs* where the segment ends, and the set of transcripts splicing the segment's region.

2.3 Yanagi-based Workflow

Figure 1 (E) gives an overview of a yanagi-based workflow which consists of three steps. The first step is the transcriptome segmentation, in which the segments library is generated. Given the transcriptome annotation and the genome sequences, and for a specific parameter value L , Yanagi generates the segments in FASTA file format. This step of library preparation is done once independently from the samples. The second step is the alignment step. Using any kmer-based aligner e.g. kallisto or RapMap, the aligner uses the segments library for library indexing and alignment. The outcome of this step is read counts per segments (in case of single-end reads) or segment-pair counts (in case of paired-end reads). These segment counts (SCs) are the statistics that yanagi provides to be used in any downstream analysis. The third step depends on the specific target analysis. Later on this work, we describe two use cases where using segment counts shows to be computationally efficient and statistically beneficial.

3 Datasets

The experiments are based on the simulation data provided by [14] for both fruit fly and human organisms (dm3 and hg37 assembly versions, respectively). Each dataset consists of samples from two conditions. Each condition has three replicates. The reads for the replicates are simulated from real RNA-seq samples, to get realistic expression values, after incorporating a variance model and the change required between conditions. The simulation is restricted to only protein-coding genes in the primary genome assembly. The difference in transcripts usage across conditions was simulated in 1000 genes randomly selected from genes with at least two transcripts and high enough expression levels. For each of these 1000 genes, the expression levels of the two most abundant transcripts is switched across conditions. Refer to [14] for full details of the preparation procedure of the dataset.

4 Analysis of Generated Segments

For practical understanding of the generated segments, we used Yanagi to build segment libraries for the fruit fly and human genomes: *Drosophila melanogaster* (UCSC dm6) and *Homo sapiens* (UCSC hg38) genome assemblies and annotations. These organisms show different genome characteristics, e.g. the fruit fly genome has longer exons and transcripts than the human genome, while the number of transcripts per gene is much higher for human genome than the fruit fly. A summary of the properties of each genome is found in [14].

4.1 Sequence lengths of generated segments

Since L is the only parameter required by the segmentation algorithm, we tried different values of L to understand the impact of that choice on the generated segments library. Recall that the choice of L is based on the expected read length of the sequencing experiment. For this analysis we chose the set $L = (40, 100, 1000, 10000)$.

Figure 2 shows the histogram of the lengths of the generated segments compared to the histogram of the transcripts lengths, for each value of L , for both fruit fly (left) and human (right) genomes. The figure shows the expected behavior when increasing the value of L ; using small values of L tends to shred the transcriptome more (higher frequencies for small sequence lengths), especially with genomes of complex splicing structure like the human genome. While with high values of L , such as $L = 10,000$, the minimum segment length anticipated tends to be higher than the length of most transcripts, ending up generating segments such that each segment represents a full transcript.

4.2 Number of generated segments per gene

Figure 3 shows how the number of generated segments in a gene is compared to the number of the transcripts in that gene, for each value of L , for both fruit fly (left) and human (right) genomes. A similar behavior is observed while increasing the value L , as with the segments length distribution. The fitted line included in each scatter plot provides indication of how the number of target sequences grows compared to the original transcriptome. For example when using $L = 100$ (a suitable value with Illumina's short reads), the number of target sequences per gene, which will be the target of the subsequent pseudo-alignment steps, almost doubles. It is clear from both figures the effect of the third step in the segmentation stage. It is important not to shred the transcriptome so much that the target sequences become very short leading to resulting complications in the pseudo-alignment and quantification steps, and not to increase the number of target sequences leading to increasing the processing complexity of these steps.

4.3 Library Size of the generated segments

As a summary, Table 1 shows the library size when using segments compared to the reference transcriptome in terms of the total number of sequences, sequence bases, and file sizes. The total number of sequence bases clearly shows the advantage of using segments to reduce repeated sequences appearing in the library that corresponds to genomic regions shared among multiple isoforms. For instance, using $L = 100$ achieves 54% and 35% compression rates in terms of sequence lengths for fruit-fly and human genomes, respectively. The higher the value of L is, the more overlap is allowed between segments, hence providing less the compression rate. Moreover, that necessarily hints on the expected behavior of the alignment step in terms of the frequency of multi-mappings.

4.4 Impact of using segments on Multi-mapped Reads

To study the impact of using the segments library instead of the transcriptome for alignment, we created segments library with different values of L and observed the number of multimapped and unmapped reads for each case and how it is compared to when the transcriptome is used. We used RapMap [6] as our kmer-based

aligner, to align samples of 40 million simulated reads of length 101 (samples from the dataset discussed in Datasets section) in a single-end mode. The experimented values of L were centered around the value of $L = 101$ with more value points close to 101 to test how sensitive the results are towards small changes in the selection of L . Figure 4 shows the alignment performance in terms of the number of multimapped reads (red solid line) and unmapped reads (blue solid line), compared to the number of multimapped reads (red dotted line) and unmapped reads (blue dotted line) when aligning using the transcriptome. Using segments highly reduces the number of multimapped reads. The plot shows that too short segments compared to the read length results in a lot of unmapped reads. Consequently, choosing L to be close to the read length is the optimal choice to minimize multimappings while maintaining a steady number of mapped reads. It is important to note that the best segments configuration still produces some multimappings. That is a result of the presence of reads sequenced from paralogs and sequence repeats that are not tackled in the current version of yanagi. However, it is clear that using segments can achieve around 10 fold decrease in the number of multimappings.

4.5 The importance of maximality property

Recalling that the generated segments are maximal segments, as mentioned in definition 2.1. It is the property that segments are extended as much as possible between branching points in the segments graph. The purpose of this property is to maintain stability in the produced segment counts; Since shorter segments will inherently produce lower counts which introduces higher variability that can complicate the downstream analysis. Figure 5 shows the distribution of coefficient of variation (CV) of the produced segment counts from segments with and without maximal property. To examine the effect of the maximal property, we simulated 10 replicates from 1000 random genes (with more than two isoforms) from the hg38 transcriptome using poyester [15]. When segments are created without maximal property, The scatter plot clearly shows that maximal segments have lower CVs to their corresponding short segments for a majority of points (40% of the points has a difference in CVs ≥ 0.05). That corresponds to generating counts with lower means and/or higher variances if the maximal property was dropped.

5 Segment-based Gene Expression Analysis

A typical segment-based approach to do gene expression analysis would start by performing kmer-based alignment over the segments library prepared earlier by Yanagi using high-throughput tools like kallisto, sailfish or RapMap, to derive segment counts (SCs). The segment counts are then used to perform differential gene expression.

The standard RNAseq pipeline for gene expression analysis depends on performing kmer-based alignment over the transcriptome to obtain transcripts abundances, e.g. transcripts per million (TPMs). Then depending on the objective of the differential analysis, an appropriate hypothesis test is used to detect genes that are differentially expressed. Methods that perform differential gene expression (DGE) prepares gene abundances by summing the underlying transcript abundances. Consequently, DGE methods aims at testing for differences in the overall gene expression. Among

these methods are: DESeq2 [16] and edgeR [17]. Such methods fails to detect cases where some transcripts switch usage levels while the total gene abundance is not significantly changing. Note that estimating gene abundances by summing counts from the underlying transcripts can be problematic, as discussed in [18]. RATs [19] on the other hand is among those methods that target to capture such behavior and tests for differential transcript usage (DTU). Regardless of the testing objective, both tests entirely depend on the transcript abundances that were obtained from algorithms like EM during the quantification step to resolve the ambiguity of the multimapped reads, which adherently requires some bias-correction modeling ([8]) adding another layer of complexity to achieve the final goal of gene analysis.

Our segment-based approach aims at breaking the coupling between the quantification, bias modeling, and gene expression analysis, while maintaining the advantage of using ultra-fast pseudo-alignment techniques provided by kmer-based aligners. When Aligning over the L-disjoint segments, the problem of multimapping across target sequences is avoided and as a result the quantification step can be dropped. Then the hypothesis test for differences across conditions are performed on SCs count matrix instead of TPMs.

5.1 Kallisto's TCC-based approach

Yi et al. introduces a comparable approach in [20]. This approach uses an intermediate set defined in kallisto's index core as equivalence classes (ECs). Specifically, a set of kmers are grouped into an equivalence class (EC) if it belongs to the same set of transcripts during the transcriptome reference indexing step. Then during the alignment step kallisto derives a count statistic for each EC. The statistics are referred to as transcripts compatibility counts (TCCs). In other words, kallisto produces one TCC per EC representing number of fragments that appeared compatible with the corresponding set of transcripts during the pseudo-alignment step. Then the work in [20] uses these TCCs to directly perform gene-level differential analysis by skipping the quantification step using logistic regression. We will refer to that direction as TCC-based approach. To put that approach into perspective with our segment-based approach, we will discuss how the two approaches are compared to each other.

5.2 Comparison between segment-based and TCC-based approaches

Both segment-based and TCC-based approaches successfully avoids the quantification step when targeting gene-level analysis. This can be seen as an advantage in efficiency, speed, simplicity, and accuracy, as previously discussed. One difference is that segment-based approach is agnostic to the alignment technique used, while TCC-based approach is a kallisto-specific approach. More importantly, the statistic used in segment-based approach is easily interpretable. Since segments are formed to preserve the genomic location and splicing structure of genes, SCs can be directly mapped and interpreted with respect to the genome coordinates. However, ECs do not have a direct biological meaning in this sense. For instance, all kmers that belong to the same transcript yet originated from different locations over the genome will all fall under the same EC, making TCCs less interpretable. While on the contrary, these kmers will appear in different segments depending on the transcriptome

structure. This advantage can be crucial for a biologist who tries to interpret the outcome of the differential analysis. In the next section we show a segment-based gene visualization that allows users to visually examine, for genes determined to be differentially expressed, what transcripts, exons and splicing events contributed to that difference.

Figure 6-bottom shows the number of yanagi's segments per gene versus the number of kallisto's equivalence classes per gene. The number of equivalence classes were obtained by building kallisto's index on hg37 transcriptome, then running the pseudo command of kallisto (kallisto 0.43) on the 6 simulated samples. Note that, in principle there should be more segments than ECs since segments preserve localization, however in practice kallisto reports more ECs than those discovered in the annotation alone in some genes. The extra ECs are formed during pseudo-alignment when reads show evidence of unannotated junctions.

5.3 DEXSeq-based model for differential analysis

In this work we adopt the DEXSeq [21] method to perform the segment-based gene differential analysis. DEXSeq is a method that performs differential exon usage (DEU). The standard DEXSeq workflow begins by aligning reads to a reference genome, not to the transcriptome, using TopHat2 or STAR [22] to derive exon counts. Then given the exon counts matrix and the transcriptome annotation, DEXSeq tests for DEU after handling coverage biases, technical and biological variations. It fits, per gene, a generalized linear model (GLM) of negative binomial (NB) accounting for effect of the condition factor, and compares it to the null model (without the condition factor) using a chi-square test. Exons that have their null hypotheses rejected are proven to be significantly different between the experimental conditions, hence DEU is achieved. DEXSeq extends its testing for DEU by controlling the false discovery rate (FDR) at gene level using the Benjamini-Hochberg procedure to find genes with at least one significantly different exon.

Adopting DEXSeq model for the case of segments is done by replacing exons with segments. In other words, the count matrix fed to DEXSeq represent segment counts, instead of exon counts. Once segments are tested for differential usage between conditions, their *p-values* are aggregated to find genes with at least one segment proven to be significantly different.

We tested that model on the simulated data for both human and fruit fly samples, and compared our segment-based approach with the TCC-based approach since they are closely comparable. Since the subject of study is the effectiveness of using either SCs or TCCs as statistic, we fed TCCs reported by kallisto to DEXSeq's model as well to eliminate any performance bias due the testing model. As anticipated, figure 6-top shows that both approaches provide highly comparable results. Furthermore, using segment counts to test for differentially expressed genes adds to the interpretability of the test outcomes. The next section shows how visualizing segment counts connects the result of the hypotheses test with the underlying biology of the gene.

6 Segment-based Gene Visualization

Figure 7 shows Yanagi's proposed method to visualize segments and the segment counts of a single gene with differentially expressed genes. The plot includes different panels combined, each showing a different aspect of the mechanisms involved in

differential expression calls. The main panel of the plot is the segment-exon membership matrix (Panel A). This matrix plot shows the structure of the segments (rows) over the exonic bins (columns) prepared during the annotation preprocessing step. Recall that an exon (or a retained intron) in the genome can be represented with more than one exonic bin in case of within-exon splicing events (Step 1 in section 2.2). Panel B is a transcript-exon membership matrix. It encapsulates the transcriptome annotation with transcripts as rows and the exonic bins as columns. Both membership matrices together allows the user to map segments (through exonic bins) to transcripts.

Panel C shows the segment counts (SCs) for each segment row. Panel D shows the length distribution of the exonic bins. Panel E is optional. It adds the transcript abundances of the samples, if provided. This can be useful to capture cases where coverage biases over the transcriptome is considered, or to capture local switching in abundances that are inconsistent with the overall abundances of the transcripts

The gene in figure 7 is on the reverse strand, that's why the exonic bins axis is reversed and segments are created from right to left. Consider segment S.0674 for instance. It was formed by spanning the first exonic bin (right-most bin) plus the junction between the first two bins. This junction is present only at transcript T.1354 and hence that segment belongs to only that transcript. In the segment-exon matrix, red-colored cells mean that the segment spans the entire bin, while salmon-colored cells represent partial bin spanning; usually at the start or end of a segment with correspondence to some junction.

Alternative splicing events can be easily visualized from figure 7. For instance, segments S.0672 and S.0671 represent an exon-skipping event where the exon is spliced in T.6733 and skipped in both T.1354 and T.9593.

7 Segment-based Alternative Splicing Analysis

Within a gene, the study of how certain genomic regions are alternatively spliced into different isoforms is related to the study of relative transcript abundances. Each local splicing event describes a possible variation of splicing of the described genomic region. For instance, an exon cassette event (exon skipping) describes either including or excluding an exon between the upstream and downstream exons. Consequently, isoforms are formed through a sequential combination of local splicing events. For binary events, the relative abundance of an event is commonly described in terms of percent spliced-in (PSI) [23] which measures the proportion of reads sequenced from one splicing possibility versus the alternative splicing possibility, while ΔPSI describes the difference in PSI across experimental conditions of interest.

Several approaches were introduced to study alternative splicing and its impact in studying multiple diseases. [24] surveyed eight different approaches that are commonly used in the area. These approaches can be roughly categorized into two categories depending on how the event abundance is derived for the analysis. The first category is considered count-based where the approach focuses on local measures spanning specific counting bins (e.g. exons or junctions) defining the event, like DEXSeq [21], MATS [25] and MAJIQ [26]. Unfortunately, many of these approaches can be expensive in terms of computation and/or storage requirements since it requires mapping reads to the genome, and then processing the huge matrix of counting bins. The second category is isoform-based where the approach

uses the relative transcript abundances as basis to derive PSI values. This direction utilizes the transcript abundance (e.g. TPMs) as a summary of the behavior of the underlying local events. Cufflinks [4, 18], DiffSplice [27] and SUPPA [28, 29] are of that category. Unlike Cufflinks and DiffSplice which perform read assembly and discovers novel events, SUPPA succeeds in overcoming the computational and storage limitations by using transcript abundances that were rapidly prepared by lightweight kmer counting alignment like Kallisto or Salmon.

A main drawback of SUPPA and other transcript-based approaches alike is that it assumes a homogeneous abundance behavior across the transcript making it prone to coverage biases. Previous work showed that RNA-seq data suffers from coverage bias that needs to be modeled into methods that estimate transcript abundances [30, 31]. Sources of bias can vary between fragment length, positional bias due to RNA degradation, and GC content in the fragment sequences. Consider the diagram in figure 8 with a case of two isoforms where isoform1 has higher abundance than isoform2. Both isoforms involve two exon skipping events (E1, E2). The diagram shows the read coverage over different regions of both isoforms with exon E1 in particular has low relative coverage. Considering the real evidence of reads supporting the first skipping event E1, gives a counter conclusion to when considering the overall abundances of the two isoforms involved. More importantly, transcript-based approaches fail to provide different measure of confidence for differential analysis of events E1 and E2 since both events will have the same *PSI* values, whereas there is a significant difference in coverage supporting both events.

Our segment-based approach works as a middle ground between count-based and transcript-based approaches. It provides local measures of splicing events while avoiding the computational and storage expenses of count-based approaches by using the rapid lightweight aligners that transcript-based approaches use. Our pipeline begins by running kmer-based lightweight alignment tools like Kallisto over the segments library prepared by Yanagi and obtain the segment counts. Yanagi's script is then used to map splicing events to their corresponding segments, e.g. each event is mapped into two sets of segments: The first set spans the inclusion splice, and the second for the alternative splice. Current version of Yanagi follows SUPPA's notation for defining a splice event and can process seven event types: Skipping Exon (SE), Retain Intron (RI), Mutually Exclusive Exons (MX), Alternative 5' splice-site (A5), Alternative 3' splice-site (A3), Alternative First Exon (AF) and Alternative Last Exon (AL).

7.1 Segment-based calculation of PSI

While Yanagi uses the transcriptome annotation to prepare the segments along with the splicing events, it generates mapping between each event and its corresponding segments spanning the event. For each event, Yanagi takes into consideration the transcripts involved and the event genomic coordinates to decide the set of transcriptome segments that correspond to each of the two possibilities of the splicing event. This step becomes complicated in case of overlapping events. The current version of Yanagi selects segments that spans either the event exon or junctions while the segment belong to at least one transcript that undergoes the corresponding splicing.

After alignment, Yanagi provides segment counts or segment-pair counts in case of paired-end reads. For each splicing event, we calculate the PSI value of event e in sample x as follows:

$$PSI(e, x) = \frac{\sum_{s \in S_i(e)} SC(s, x)}{\sum_{s \in S_i(e) \cup S_e(e)} SC(s, x)}$$

where $S_i(e)$ and $S_e(e)$ are inclusion and exclusion segments, respectively, and $SC(s, x)$ is the segment count in the sample. That means segment-based PSI values uses reads spanning both the junctions and the target inclusion exon towards the inclusion count. In fact, read counts will also include reads extended around the event as long as the segment extends on both sides. This extension takes advantage of situations where splicing events are near to include as much discriminative reads into the counts to achieve higher levels of confidence when calculating PSI values.

7.2 PSI comparison on simulated data

we compared PSI values obtained from our approach versus counting-based approaches like rMATS and isoform-based approaches like SUPPA2 on splicing events found in hg37 based on the six samples in section 3. Since each tool provides different set of events, We focus our comparison on the intersection set of events between SUPPA and rMATS. That includes events from five types of splicing events. Table 3 summarizes the number of events subject to the study. Two levels of filtering are applied to observe how the different approaches behave in different scenarios. Non-overlapping events is the smallest subset of events where there is no more splicing other than the two possibilities defining the event, i.e. complex splicing is excluded. While highTPM events is a subset of events in which inclusion and exclusion isoform levels are relatively high ($TPM_{inc} > 1, TPM_{ex} > 1$). This is a typical filtering criteria adopted by isoform-based approaches. This filter excludes events involving isoforms of low levels of expression which inherently suffer from low estimation accuracy.

Figure 9 shows a scatter plot of PSI values calculated by the three approaches. It is clear that our segment counts (SCs) based approach produces results comparable to rMATS with average Pearson correlation of 0.92 over the full set of events. As expected, PSI values obtained by our approach and rMATS are more correlated to each other than to values derived directly from TPMs, since both our approach and rMATS's are counts based.

Our results and rMATS are consistently comparable across the three subsets of events. In other words, both approaches give comparable results for cases of events with complex overlapping splicing, While results start to diverge from isoform-based results for overlapping events. On the other hand, results from isoform-based start to be less correlated with the other two approaches when events with low TPMs are included.

Among the five different splicing types exon skipping, alternative 3' and alternative 5' events gives the highest correlation between segment counts and rMATS approaches. In our experiments we noticed that rMATS (v4.0.1) does not behave as intended for intron retention events. We noticed that counts including junction

reads only and counts including both junction and intron reads (which we use in this study) are the same. In other words, rMATS fails to report reads spanning the intron, which explains the underestimated inclusion counts and PSI values for retained introns.

7.3 Differential Alternative Splicing

Since the scope of this paper is to introduce the use of segment counts as a statistic for studying alternative splicing, we want to use the simplest statistical model for differential splicing to exclude any advantage of the model itself. In that matter we used the PSI values of the three approaches (SCs, rMATS, TPM) as discussed in the previous section. Then we used a linear model for differential hypothesis testing (implemented with Limma-voom R Package [32, 33]). However, more advanced models of differential analysis can be used instead. For example, a similar model to SUPPA2 can be developed to test the significance of ΔPSI by considering all events genome-wide [29]. Figure 10 shows ROC plots for sensitivity and specificity measures. Using segment counts achieves comparable performance to both rMATS and isoform-based approaches.

8 Discussion

In this paper, we formalized the concept of transcriptome segmentation and propose an efficient algorithm for generating segment libraries based on a length parameter dependent on specific RNA-Seq library construction. The resulting segment sequences were used with pseudo-alignment tools to quantify expression at the segment level. We characterized the segment libraries for the reference transcriptomes of *Drosophila melanogaster* and *Homo sapiens* and provided gene-level visualization of the segments for better interpretability. We demonstrated the use of segments-level quantification into gene expression and alternative splicing analysis. The notion of transcript segmentation as introduced here and implemented in Yanagi opens the door for the application of lightweight, ultra-fast pseudo-alignment algorithms in a wide variety of RNA-seq analyses.

Using segment library rather than the standard transcriptome succeeds in significantly reducing ambiguous alignments where reads are multimapped to several sequences in the reference. That allowed avoiding the quantification step required by standard kmer-based pipelines for gene expression analysis. Moreover, using segment counts as statistics for alternative splicing analysis enables achieving comparable performance to counting-based approaches (e.g. rMATS) while rather using fast and lightweight pseudo alignment that are several folds faster than standard counting pipelines.

Competing interests

The authors declare that they have no competing interests.

Author's contributions

HCB, SMM, and MKG conceived and designed the study. HCB and MKG designed all software tools used in the study, MKG implemented all software tools used in the study. MKG acquired all data used and analyzed and interpreted results with HCB. MKG wrote the initial draft of the manuscript, HCB and SMM participated the final drafting of the manuscript.

Acknowledgements

We want to thank Julien Buchbinder and Steffen Cornwell for preliminary work in transcript segmentation. This work was partially supported by NSF grant ABI 1564785 to SMM and MKG, and NIH grants HG005220 and GM114267 to HCB and MKG.

Author details

¹Department of Computer Science, University of Maryland, College Park, Maryland, USA. ²Center for Bioinformatics and Computational Biology, University of Maryland, College Park, Maryland, USA. ³Department of Cell Biology and Molecular Genetics, University of Maryland, College Park, Maryland, USA.

References

1. Langmead, B., Salzberg, S.L.: Fast gapped-read alignment with bowtie 2. *Nature methods* **9**(4), 357–359 (2012)
2. Trapnell, C., Pachter, L., Salzberg, S.L.: Tophat: discovering splice junctions with rna-seq. *Bioinformatics* **25**(9), 1105–1111 (2009)
3. Kim, D., Pertea, G., Trapnell, C., Pimentel, H., Kelley, R., Salzberg, S.L.: Tophat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome biology* **14**(4), 36 (2013)
4. Trapnell, C., Williams, B.A., Pertea, G., Mortazavi, A., Kwan, G., Van Baren, M.J., Salzberg, S.L., Wold, B.J., Pachter, L.: Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology* **28**(5), 511–515 (2010)
5. Patro, R., Mount, S.M., Kingsford, C.: Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nature biotechnology* **32**(5), 462–464 (2014)
6. Srivastava, A., Sarkar, H., Gupta, N., Patro, R.: Rapmap: a rapid, sensitive and accurate tool for mapping rna-seq reads to transcriptomes. *Bioinformatics* **32**(12), 192 (2016)
7. Bray, N.L., Pimentel, H., Melsted, P., Pachter, L.: Near-optimal probabilistic rna-seq quantification. *Nature biotechnology* **34**(5), 525–527 (2016)
8. Patro, R., Duggal, G., Love, M.I., Irizarry, R.A., Kingsford, C.: Salmon provides fast and bias-aware quantification of transcript expression. *Nature Methods* (2017)
9. Li, B., Dewey, C.N.: Rsem: accurate transcript quantification from rna-seq data with or without a reference genome. *BMC bioinformatics* **12**(1), 323 (2011)
10. Zipursky, S.L., Wojtowicz, W.M., Hattori, D.: Got diversity? wiring the fly brain with dscam. *Trends in biochemical sciences* **31**(10), 581–588 (2006)
11. Tilgner, H., Jahanbani, F., Blauwkamp, T., Moshrefi, A., Jaeger, E., Chen, F., Harel, I., Bustamante, C.D., Rasmussen, M., Snyder, M.P.: Comprehensive transcriptome analysis using synthetic long-read sequencing reveals molecular co-association of distant splicing events. *Nature biotechnology* **33**(7), 736–742 (2015)
12. Haas, B.J., Delcher, A.L., Mount, S.M., Wortman, J.R., Smith Jr, R.K., Hannick, L.I., Maiti, R., Ronning, C.M., Rusch, D.B., Town, C.D., *et al.*: Improving the arabidopsis genome annotation using maximal transcript alignment assemblies. *Nucleic acids research* **31**(19), 5654–5666 (2003)
13. Gunady, M.K., Cornwell, S., Mount, S.M., Bravo, H.C.: Yanagi: Transcript Segment Library Construction for RNA-Seq Quantification. In: Schwartz, R., Reinert, K. (eds.) 17th International Workshop on Algorithms in Bioinformatics (WABI 2017). Leibniz International Proceedings in Informatics (LIPIcs), vol. 88, pp. 10–11014. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2017). doi:10.4230/LIPIcs.WABI.2017.10. <http://drops.dagstuhl.de/opus/volltexte/2017/7648>
14. Sonesson, C., Matthes, K.L., Nowicka, M., Law, C.W., Robinson, M.D.: Isoform prefiltering improves performance of count-based methods for analysis of differential transcript usage. *Genome biology* **17**(1), 12 (2016)
15. Frazee, A.C., Jaffe, A.E., Langmead, B., Leek, J.T.: Polyester: simulating rna-seq datasets with differential transcript expression. *Bioinformatics* **31**(17), 2778–2784 (2015)
16. Love, M.I., Huber, W., Anders, S.: Moderated estimation of fold change and dispersion for rna-seq data with *deseq2*. *Genome biology* **15**(12), 550 (2014)
17. McCarthy, D.J., Chen, Y., Smyth, G.K.: Differential expression analysis of multifactor rna-seq experiments with respect to biological variation. *Nucleic Acids Research* **40**(10), 4288–4297 (2012). doi:10.1093/nar/gks042
18. Trapnell, C., Hendrickson, D.G., Sauvageau, M., Goff, L., Rinn, J.L., Pachter, L.: Differential analysis of gene regulation at transcript resolution with rna-seq. *Nature biotechnology* **31**(1), 46 (2013)
19. Froussios, K., Mourão, K., Schurch, N.J., Barton, G.J.: Identifying differential isoform abundance with rats: a universal tool and a warning. *bioRxiv* (2017). doi:10.1101/132761. <https://www.biorxiv.org/content/early/2017/05/01/132761.full.pdf>
20. Yi, L., Pimentel, H., Bray, N.L., Pachter, L.: Gene-level differential analysis at transcript-level resolution. *bioRxiv*, 190199 (2017)
21. Anders, S., Reyes, A., Huber, W.: Detecting differential usage of exons from rna-seq data. *Genome research* **22**(10), 2008–2017 (2012)
22. Dobin, A., Davis, C.A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., Gingeras, T.R.: Star: ultrafast universal rna-seq aligner. *Bioinformatics* **29**(1), 15–21 (2013)
23. Schafer, S., Miao, K., Benson, C.C., Heinig, M., Cook, S.A., Hubner, N.: Alternative splicing signatures in rna-seq data: Percent spliced in (psi). *Current protocols in human genetics*, 11–16 (2015)
24. Liu, R., Loraine, A.E., Dickerson, J.A.: Comparisons of computational methods for differential alternative splicing detection using rna-seq in plant systems. *BMC bioinformatics* **15**(1), 364 (2014)
25. Shen, S., Park, J.W., Lu, Z.-x., Lin, L., Henry, M.D., Wu, Y.N., Zhou, Q., Xing, Y.: rmat: robust and flexible detection of differential alternative splicing from replicate rna-seq data. *Proceedings of the National Academy of Sciences* **111**(51), 5593–5601 (2014)
26. Vaquero-Garcia, J., Barrera, A., Gazzara, M.R., Gonzalez-Vallinas, J., Lahens, N.F., Hogenesch, J.B., Lynch, K.W., Barash, Y., Valcárcel, J.: A new view of transcriptome complexity and regulation through the lens of local splicing variations. *eLife* **5**, 11752 (2016)
27. Hu, Y., Huang, Y., Du, Y., Orellana, C.F., Singh, D., Johnson, A.R., Monroy, A., Kuan, P.-F., Hammond, S.M., Makowski, L., *et al.*: Diffsplice: the genome-wide detection of differential splicing events with rna-seq. *Nucleic acids research* **41**(2), 39–39 (2012)

28. Alamancos, G.P., Pagès, A., Trincado, J.L., Bellora, N., Eyras, E.: Suppa: a super-fast pipeline for alternative splicing analysis from rna-seq. *bioRxiv*, 008763 (2014)
29. Trincado, J.L., Entizne, J.C., Hysenaj, G., Singh, B., Skalic, M., Elliott, D.J., Eyras, E.: Suppa2 provides fast, accurate, and uncertainty-aware differential splicing analysis across multiple conditions. *bioRxiv*, 086876 (2017)
30. Lahens, N.F., Kavakli, I.H., Zhang, R., Hayer, K., Black, M.B., Dueck, H., Pizarro, A., Kim, J., Irizarry, R., Thomas, R.S., *et al.*: Ivt-seq reveals extreme bias in rna sequencing. *Genome biology* **15**(6), 86 (2014)
31. Love, M.I., Hogenesch, J.B., Irizarry, R.A.: Modeling of rna-seq fragment sequence bias reduces systematic errors in transcript abundance estimation. *Nature biotechnology* **34**(12), 1287 (2016)
32. Smyth, G.K., *et al.*: Linear models and empirical bayes methods for assessing differential expression in microarray experiments. *Stat Appl Genet Mol Biol* **3**(1), 3 (2004)
33. Law, C.W., Chen, Y., Shi, W., Smyth, G.K.: Voom: precision weights unlock linear model analysis tools for rna-seq read counts. *Genome biology* **15**(2), 29 (2014)

Figures

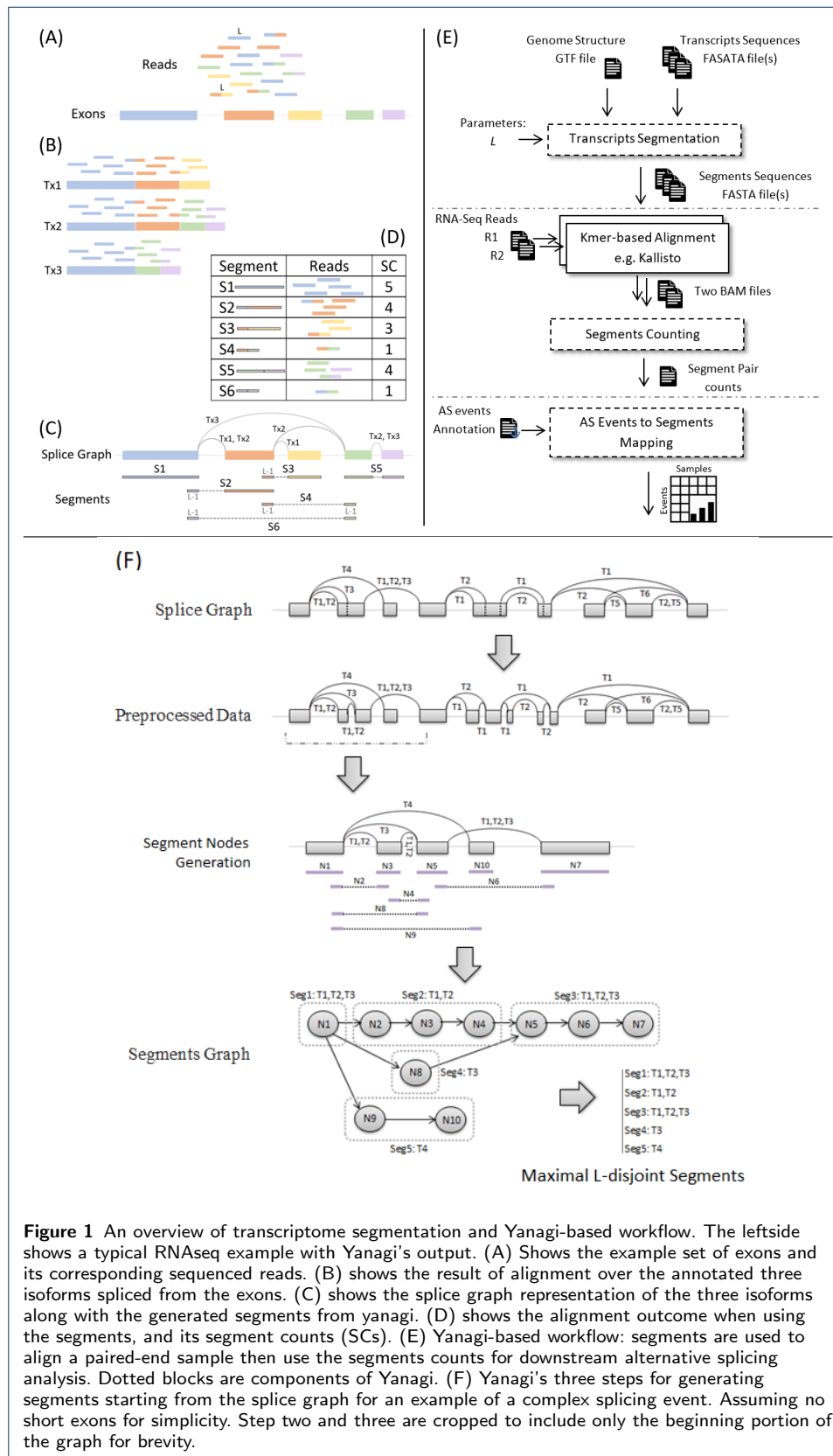
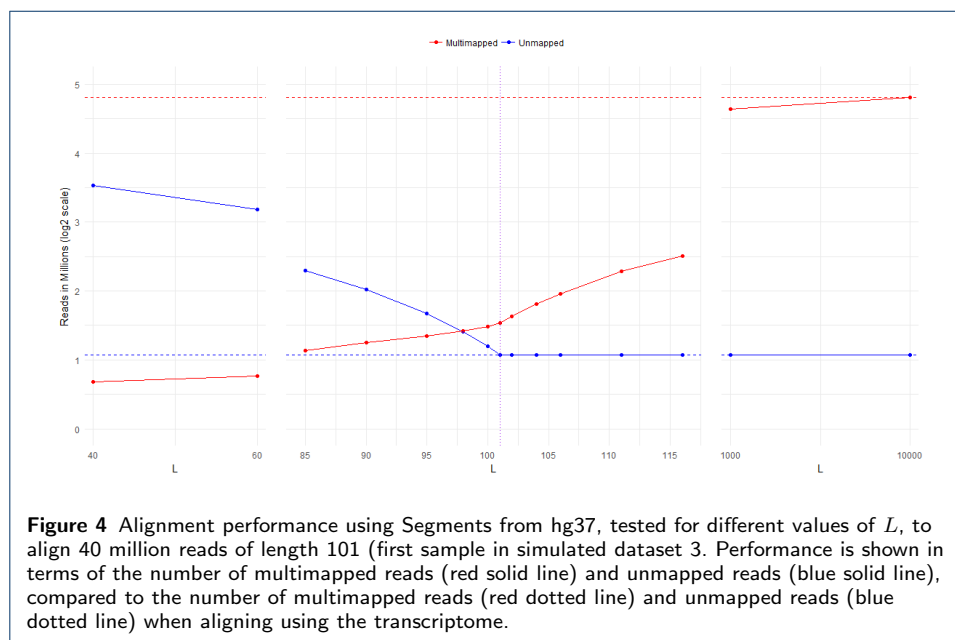
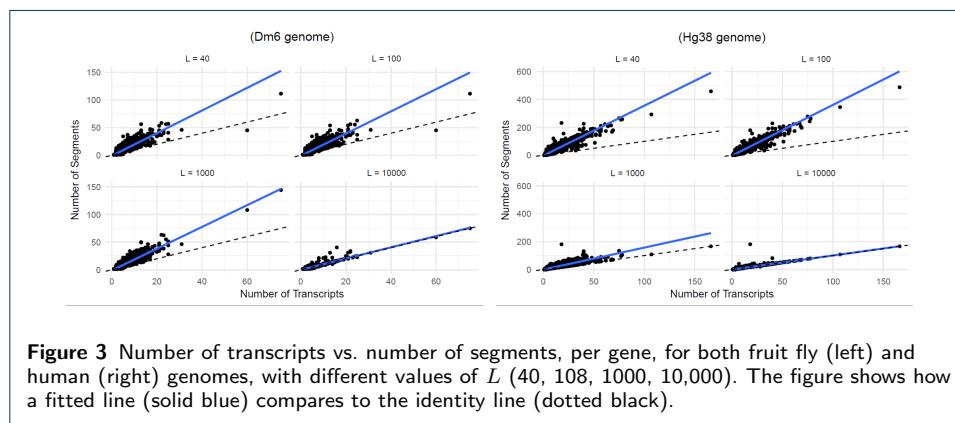
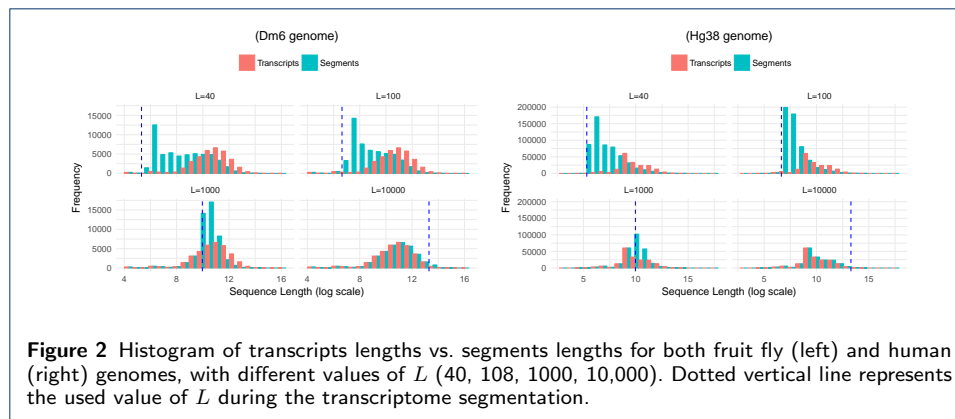
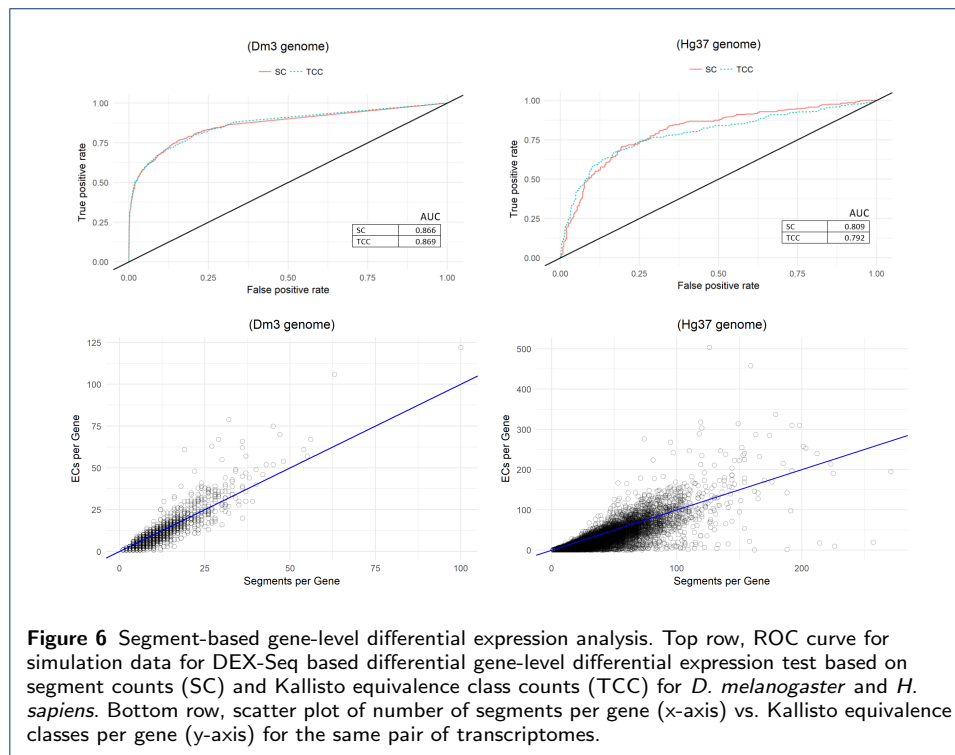
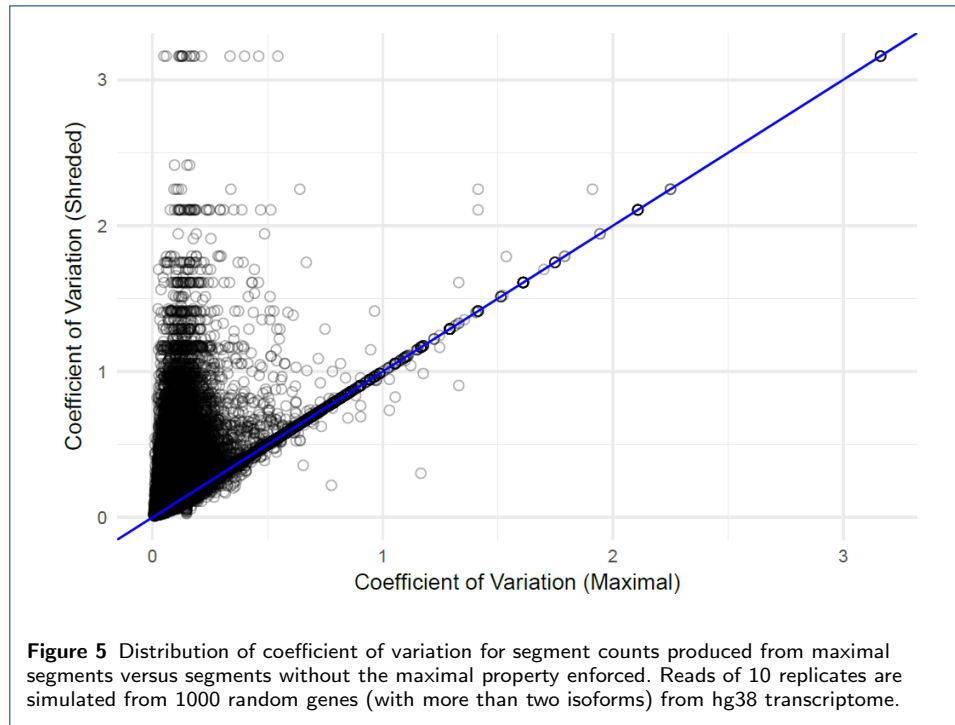
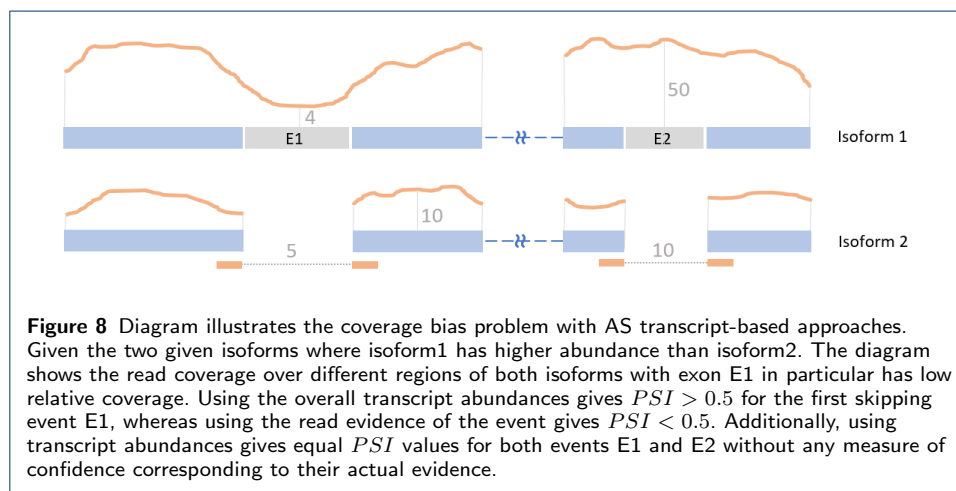
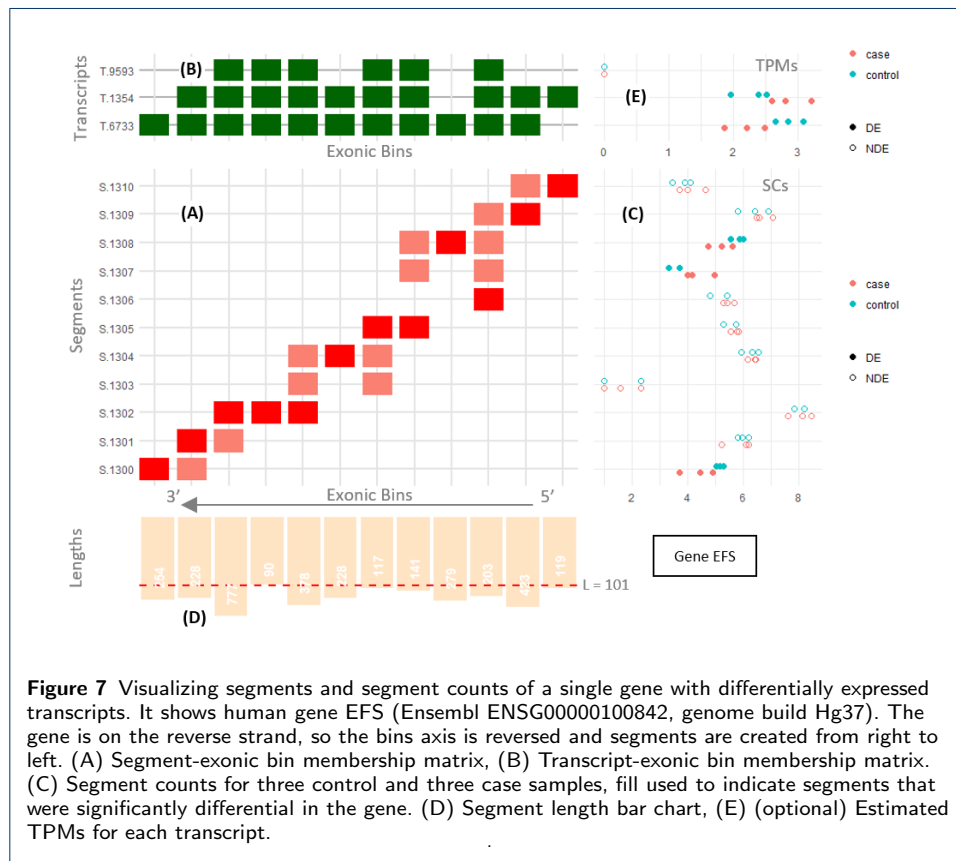


Figure 1 An overview of transcriptome segmentation and Yanagi-based workflow. The leftside shows a typical RNAseq example with Yanagi's output. (A) Shows the example set of exons and its corresponding sequenced reads. (B) shows the result of alignment over the annotated three isoforms spliced from the exons. (C) shows the splice graph representation of the three isoforms along with the generated segments from yanagi. (D) shows the alignment outcome when using the segments, and its segment counts (SCs). (E) Yanagi-based workflow: segments are used to align a paired-end sample then use the segments counts for downstream alternative splicing analysis. Dotted blocks are components of Yanagi. (F) Yanagi's three steps for generating segments starting from the splice graph for an example of a complex splicing event. Assuming no short exons for simplicity. Step two and three are cropped to include only the beginning portion of the graph for brevity.







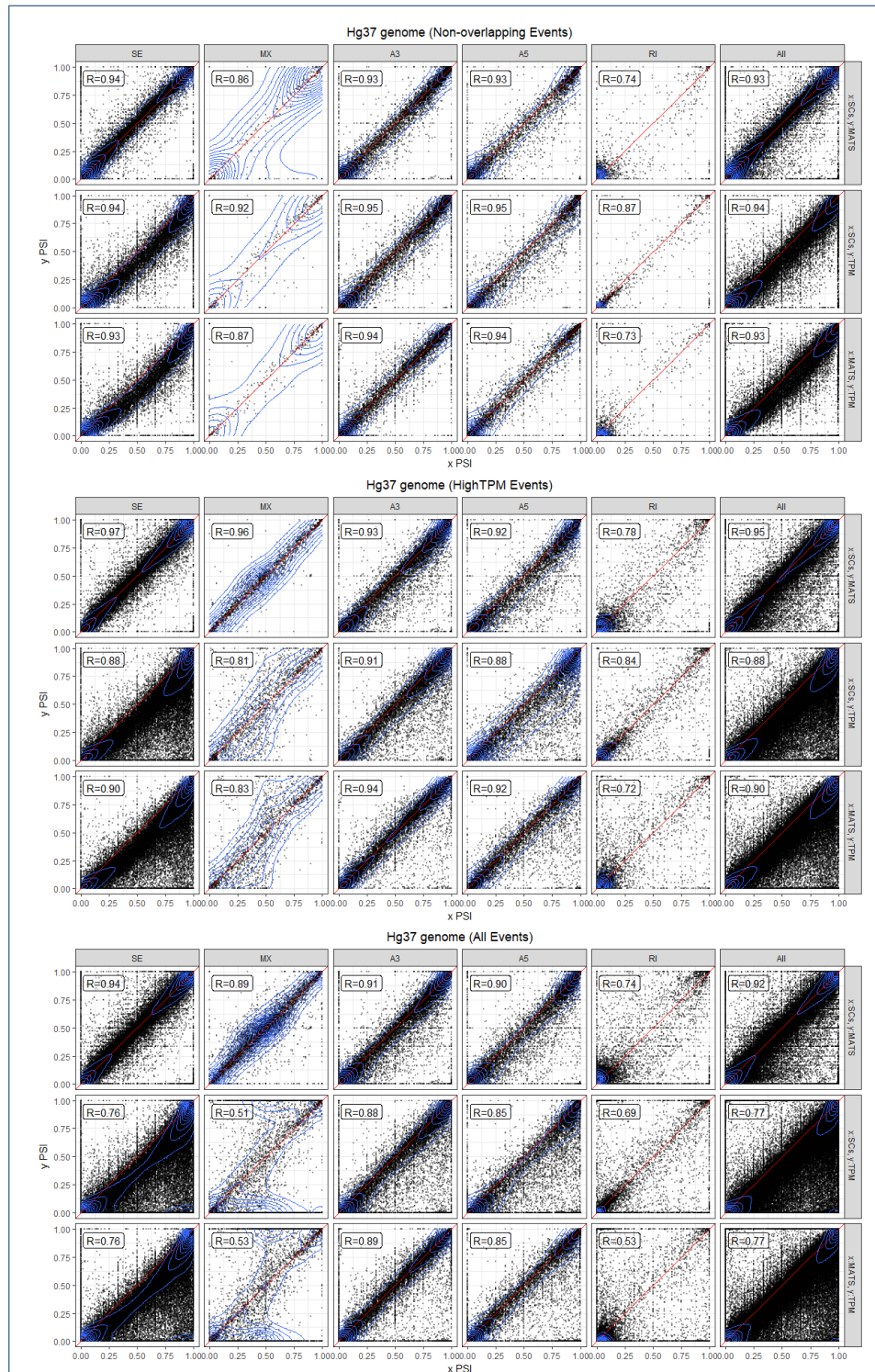
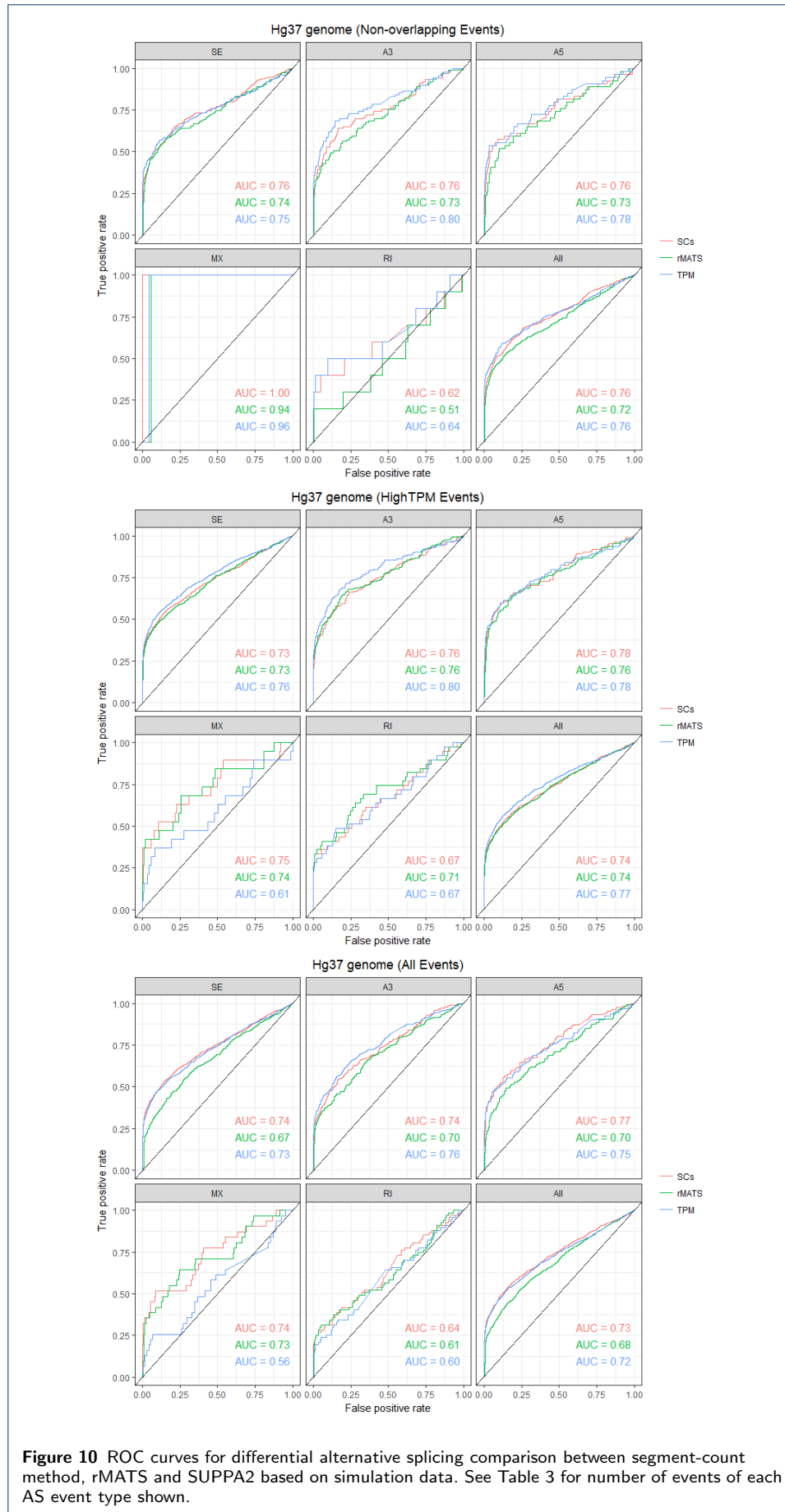


Figure 9 PSI value comparison between segment counts, rMATS (based on spliced alignment to genome) and SUPPA2 (based on estimated TPMs from pseudo-alignment and quantification). Columns indicate seven types of alternative splicing events. Scatterplots are stratified by event types (non-overlapping, high TPM, and all events). See Table 3 for number of events of each AS event type shown.



Tables

Table 1 Library size summary

	Transcriptome	Segments			
		$L = 40$	$L = 100$	$L = 1000$	$L = 10000$
Dm6					
Number of bases (Gb)	90	39	41	71	90
Number of Sequences	34,681	54,680	53,694	48,741	34,625
FASTA File Size (MB)	89	44	47	76	92
Hg38					
Number of bases (Gb)	278	147	181	308	281
Number of Sequences	182,435	544,991	541,361	264,083	183,165
FASTA File Size (MB)	276	206	239	338	302

Additional Files

Additional file 1 — Sample additional file title

Additional file descriptions text (including details of how to view the file, if it is in a non-standard format or the file extension). This might refer to a multi-page table or a figure.

Additional file 2 — Sample additional file title

Additional file descriptions text.

Table 2 Running time (seconds) and memory usage (gigabytes) by Yanagi to generate segment library for fruit fly (Dm6) and human (Hg38) genomes, for both the preprocessing and segmentation steps. Time for the preprocessing step does not include the time to load the FASTA and GTF files. Most of the memory usage is from loading the input data in both steps. Running on a 6-core 2.1 GHz AMD processor, using single-threaded processes. The lower half shows the time and memory usage for running Rapmap's quasi-mapping using the segments library and the the full transcriptome, to quantify samples of 40M paired-end reads, each of length 101bp.

	Dm6		Hg38	
	time(s)	memory(GB)	time(s)	memory(GB)
Preprocessing	13	0.9	112	1.5
Segmentation				
L=40	20	0.4	248	1.3
L=108	20	0.4	250	1.3
L=1000	20	0.4	228	1.3
L=10000	8.5	0.4	77	1.3
Rapmap Indexing (4 Threads)				
L=108	103	0.8	420	2.6
Txs	121	1.1	480	3.7
Rapmap Quantification (8 Threads)				
L=108	236	0.7	220	2.1
Txs	292	1.2	416	3.1

Table 3 Number of Events in Hg37 common between MATS and SUPPA for the five event types reported by both tools. Two levels of filtering are applied to obtain three subsets. Non-overlapping events are the simplest events where there is no more splicing other than the two possibilities defining the event. While highTPM events are events where inclusion and exclusion isoform levels are relatively high ($TPM_{inc} > 1, TPM_{ex} > 1$).

Events Subset	SE	MX	A3	A5	RI	Total
Non-overlapping	4,180	68	1,435	885	323	6,891
HighTPM Events	9,756	354	2,327	1,483	793	14,713
All Events	13,650	1,024	3,131	2,053	1,711	21,569