

Shared and distinct exonic parts in alternative paths of splicing bubbles.

Daniel Witoslawski¹, Jelard Aquino¹, Chuanchuan He¹, Mira V. Han^{1,*}

¹School of Life Sciences, University of Nevada, Las Vegas, NV 89154.

*To whom correspondence should be addressed.

Abstract

Alternative splicing creates complex “bubbles” in splicing graphs where more than two transcript paths compete, challenging methods designed for simple binary events. We present a unified framework that compares paths using distinct exonic parts observed directly from reads. We build a GrASE splicing graph (DAG) per gene, enumerate bubbles, and quantify shared and distinct exonic parts across three comparison structures: (i) all-pairwise contrasts; (ii) a multinomial n-way comparison and (iii) valid bipartitions of paths. For (iii) we introduce lower-set bipartitioning, which respects subset relations among paths by enumerating downward-closed sets in a containment graph, yielding valid two-group splits with nonempty distinguishing parts. Our test statistic is the fraction of reads mapped to distinct parts relative to distinct + shared parts, enabling differential usage across samples. The lower-set enumeration runs in $O((n + m) L(g))$ time, typically far below exhaustive powerset search. Applied to genome annotations, the approach examines more bubbles than prior tools while remaining tractable and interpretable.

Introduction

Alternative splicing (AS) enables different RNA isoforms to be generated from the same gene, expanding the functional capacity of a finite genome. AS can affect the mRNA stability, localization or translation, or can lead to different protein isoforms with diverse functions. Several computational methods have been developed to detect AS from RNA-seq data. The splice graph, a Directed Acyclic Graph (DAG) that represents the splicing patterns of a gene was important in the early formulation of the problem and allowed the development of efficient solutions (Heber *et al.*, 2002; Xing *et al.*, 2004). In this framework, the complete set of transcripts can be represented by the possible paths along the DAG, and local AS events can be defined by the bubbles in the graph that represent alternative paths between two nodes. Nested bubbles can lead to structures with multiple alternative paths that quickly increase in complexity. So, the initial methods focused on simple localized events that have only two alternative paths that are easily classified and quantified. For example, rMATS (Shen *et al.*, 2014), MISO (Katz *et al.*, 2010) or SUPPA2 (Trincado *et al.*, 2018), while widely used, are restricted to specific binary events, such as skipped exons (SE), alternative 5'/3' splice sites (A5SS, A3SS), mutually exclusive exons (MXE) and retained introns (RI), but do not support multi-path or complex events.

Several tools have since been developed to overcome these limitations and expand the detection to complex events. Tools like JUM (Wang and Rio, 2018) or MAJIQ (Vaquero-Garcia *et al.*, 2023) approached the problem by focusing on each donor or acceptor independently, instead of looking at the whole bubble. *AS sub-structures* in JUM or *local splicing variations* in MAJIQ are similar concepts that are defined as sets of splice junctions sharing a common donor or a common acceptor. This allowed them to detect and quantify complex, multi-junction events, but they gave up on resolving the paths from the donor to the acceptor. Whippet (Sterne-Weiler *et al.*, 2018) approached the problem by focusing on the local bubbles surrounding target nodes (exonic parts) that are limited by a span – defined by the farthest upstream and downstream nodes that are

directly connected to the target node. This approach elegantly defines a localized complex AS event, but the detection relies on quantifying individual sub-transcript paths that cannot be directly observed, so Expectation-Maximization is used for this purpose. The common strategy among these solutions is to include complex events with more than two alternative paths, but limit the boundaries of the local subgraphs to prevent an explosion in complexity.

Here, we propose a generalized approach to complex splicing events that attempts to globally examine all bubbles in the graph. We sort all bubbles in the graph in partial order based on hierarchy, and examine them up to a threshold size, while collapsing the internal bubbles as we move up the hierarchy. Then, to generate the comparisons among the n possible paths (the alternatives in alternative splicing), we consider three different approaches. i) all the n choose 2 combination of n paths ii) each path as a multinomial outcome iii) valid bipartition of n paths. We find AS events by identifying shared and distinct parts in each of these comparisons, quantifying these parts, and comparing them across samples. We show that although we are forced to skip larger bubbles in the graphs, the approach examines more AS bubbles compared to existing methods. Detection relies on the quantification of shared and distinct exonic parts instead of sub-transcript paths, which obviates the need for estimation and uses direct observation of mapped reads.

Structuring Isoform Comparisons

We introduced a splicing graph, that we called GrASE (Aquino *et al.*, 2025), that follows the convention as implemented in the SplicingGraph package in Bioconductor (Bindreither *et al.*, 2022), but has been extended to include the exonic part edges along the genome (Aquino *et al.*, 2025). GrASE is a directed acyclic graph (DAG) where vertices (nodes) represent the splicing sites for a given gene, ordered by their position from 5' to 3'. There are three types of edges, exons, introns, and exonic parts that connect the splicing sites, with orientations following the 5' to 3' direction. Our splicing graphs are constructed for each gene separately, based on reference gene annotations from Gencode (ver 34).

After constructing the splicing graph, we identify alternative splicing as the complete set of bubbles based on the gene annotations. Valid bubbles are structures in the graph where there exist at least two distinct valid paths – a path followed by at least 1 annotated transcript – between two nodes of a splicing graph. The nodes at the ends of the bubble are called “source” and “sink”, and the number of distinct valid paths is the “size” of the bubble. Bubbles can be nested or overlapping without being nested.

Once we identify the set of AS events as bubbles, we need to decide how to structure the comparison among the multiple alternative paths so we can compare the read coverage. JUM or MAJIQ used a multinomial strategy where they quantify and compare one out of multiple outcomes. Whippet’s approach focusing on a target node naturally leads to an intuitive partitioning of multiple paths into two clusters—inclusion paths vs. exclusion paths based on the inclusion of the target node. In this paper, we propose three different ways to look at the problem.

(i) Pairwise $\binom{n}{2}$ comparisons - We compare every pair of individual paths, generating $\binom{n}{2} = n(n - 1)/2$ pairwise comparisons. This approach provides the most granular analysis, revealing all pairwise differences between individual isoforms, and enables construction of distance matrices

and similarity networks for transcript clustering. However, it scales quadratically with the number of paths and does not leverage biological groupings.

(ii) multinomial comparisons - We organize the n paths into n singleton groups, treating each path as a distinct outcome of the alternative splicing event. This yields exactly n outputs (one per path), which we evaluate jointly in a single multinomial analysis, making it more efficient than enumerating all pairs. A limitation is that this one-vs-rest strategy will yield many paths lacking path-unique exonic parts, as we describe in the next section. If path unique parts are missing for even a single path, the multinomial comparison as defined here is not applicable.

(iii) bipartition (two-group split) comparisons - We partition the n paths into exactly two groups yielding differential exonic parts by enumerating the valid bipartitions of the multiple alternative paths. Note that the number of nontrivial bipartitions grows exponentially as $2^{n-1} - 1$, so constraints on validity are helpful for tractability. We describe what valid bipartitions mean in more detail below.

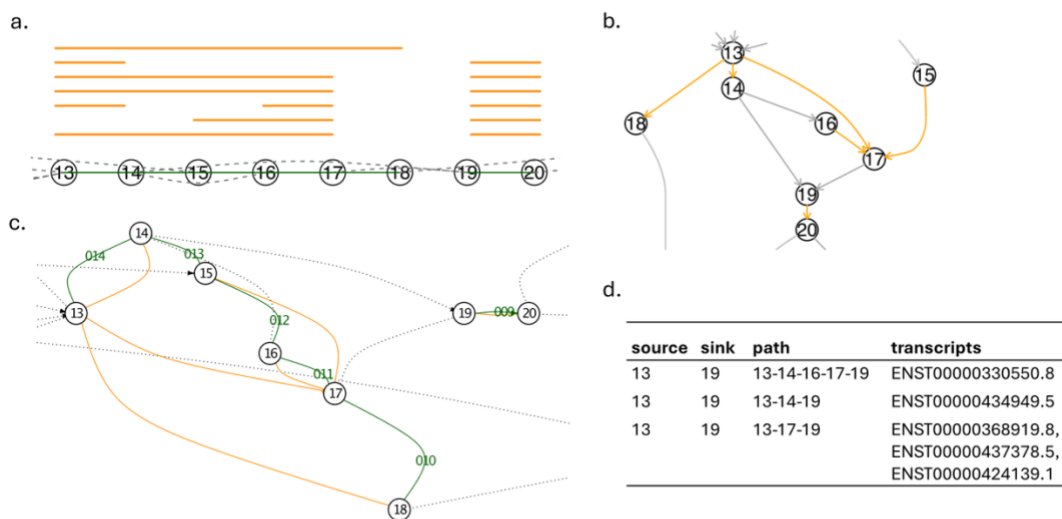
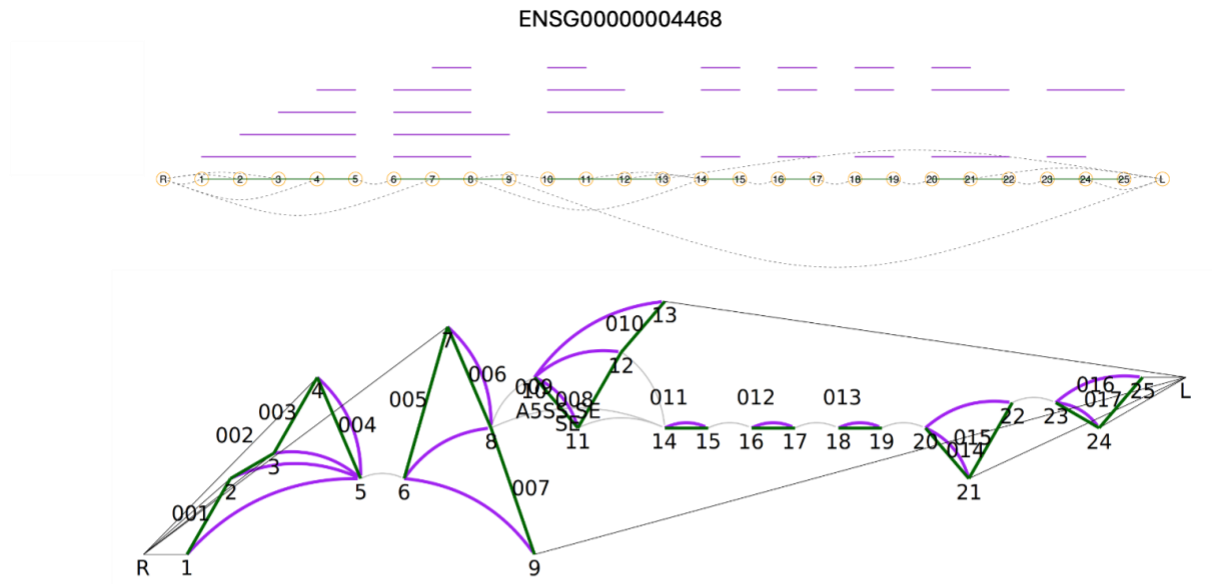


Figure 1 Example of a complex bubble. Describes the region in gene ENSG0000004809 where a bubble with 3 alternative paths is found between node 13 and 19 a) partial transcript structure b) splicing graph c) GrASE graph d) alternative paths and corresponding transcripts.

Lower set bipartitioning of paths

With complex bubbles that have n paths, there are potentially $2^{n-1} - 1$ ways to partition the alternative paths into binary classes. One may think that one intuitive way to partition the paths is a one-vs-rest (OVR) strategy, where each path is split from the rest into two partitions. This strategy is frequently used to transform a multinomial (multi class) problem to a bipartition (binary class) problem in biological contexts, e.g. Is this a T-cell or not? Does this species exist in this metagenome or not? Unfortunately, OVR partitioning is not always meaningful in the case of



bubble R-5

Alternative paths constituting the bubble R-5

vertex path	exon path	exonic part path
R-1-5	$intron_{R,1}-exon_{1,5}$	$intron_{R,1}-ex_part_{001}-ex_part_{002}-ex_part_{003}-ex_part_{004}$
R-2-5	$intron_{R,2}-exon_{2,5}$	$intron_{R,2}-ex_part_{002}-ex_part_{003}-ex_part_{004}$
R-3-5	$intron_{R,3}-exon_{3,5}$	$intron_{R,3}-ex_part_{003}-ex_part_{004}$
R-4-5	$intron_{R,4}-exon_{4,5}$	$intron_{R,4}-ex_part_{004}$

Exonic parts in one-vs-rest bipartitioning of paths

Partition 1	Partition 2	distinct exonic part
{001,002,003,004}	{002,003,004},{003,004},{004}	001
{002,003,004}	{001,002,003,004},{003,004},{004}	NA
{003,004}	{001,002,003,004},{002,003,004},{004}	NA
{004}	{001,002,003,004},{002,003,004},{003,004}	003

Exonic parts in lower set bipartitioning of paths

Partition 1	Partition 2	distinct exonic part
{001,002,003,004}	{002,003,004},{003,004},{004}	001
{001,002,003,004},{002,003,004}	{003,004},{004}	002
{001,002,003,004},{002,003,004},{003,004}	{004}	003

Figure 2. Example of the lower set bipartition. Shows the case in bubble R-5 of gene ENSG00000004809 where one-vs-rest partitioning results in empty sets for distinct exonic parts. Lower set bipartitioning results in meaningful partitions with distinct exonic parts representing the difference between the partitions.

partitioning alternative splicing paths due to their nested relationships. The purpose of the path partitioning here is to identify exonic parts that are distinct between the two group of alternative paths so we can compare their usage. The paths can be represented as sets of exonic part edges by dropping the intron edges which are irrelevant. The distinct exonic parts are edges that are present in all sets within one partition and entirely absent from the other partition. But, as seen in Figure 2, if one partition contains both the supersets and the subsets of the paths present in the other partition, as in certain OVR partitioning, then no such distinct exonic parts will be available.

To find meaningful partitions that lead to distinct exonic parts, we need to find partitions that respect the subset relationships among the paths: if a set is present in a partition, all the subsets are also present in the same partition. To solve this, we define an order among the paths in a bubble based on the subset relationship (*i.e.* subset to superset order). We then enumerate all proper downward-closed subsets (lower sets) and return each as a valid binary partition: the lower set and its compliment. Instead of exhaustively listing all powersets of the paths and checking its downward closedness, we used a more efficient algorithm described below that systematically generates all the lower sets through recursion along the containment graph.

Input: dag = containment graph with paths as nodes and edges connecting subset $\rightarrow$ superset
Output: list of valid partitions (downward-closed subsets)

nodes = all path names in dag
n = number of nodes

topo = topological sort of dag (subsets come before supersets)

For each node v in topo:
preds[v] = list of direct predecessors of v (nodes with edges to v)

Initialize empty list valid_splits

Define recursive function recurse(i, current_set):

If $i > n$:

If current_set is non-empty and not full:

Add current_set and its complement to valid_splits

Return

Shared and distinct components in comparison of alternative splicing paths.

In the simple binary AS events defined as A3SS, A5SS, SE, or RI, there are inclusion and exclusion reads that are counted separately. We can generalize these concepts to the exonic parts of complex events by identifying the exonic parts that are distinct between the comparisons, and exonic parts that are shared between the comparisons. This definition applies to all the comparison structures described in Section X. The shared exonic parts between the comparisons are analogous to the constitutive region in simple binary events, *i.e.* the exonic parts that are the same in the inclusion & exclusion isoforms. The distinct exonic parts between comparisons are analogous to the difference that defines inclusion vs. exclusion in the simple binary events. Once

these are identified, the ratio of the reads mapping to distinct parts over the reads mapping to distinct and shared parts will be the statistic that is tested for differential usage.

To identify the distinct and shared exonic parts between comparisons, we follow the logic below:
 $partition1 = \{P_1, P_2, \dots, P_m\}$, $partition2 = \{Q_1, Q_2, \dots, Q_n\}$, where P_i and Q_j represent sets of exonic part edges for each path in the bubble.

$$I_1 = \bigcap_{i=1}^m P_i, U_1 = \bigcup_{i=1}^m P_i$$

$$I_2 = \bigcap_{j=1}^n Q_j, U_2 = \bigcup_{j=1}^n Q_j$$

I and U are the intersection and the union of paths in each partition.

$$D_1 = \bigcap_{i=1}^m P_i - U_2$$

$$D_2 = \bigcap_{j=1}^n Q_j - U_1$$

D_1 and D_2 captures the distinct exonic parts that are exclusively shared by all paths in one partition but not the other partition.

$$S = I_1 \cap I_2 = \bigcap_{i=1}^m P_i \cap \bigcap_{j=1}^n Q_j$$

S captures the shared exonic parts that are universally common in all paths of both partitions. In addition to the exonic parts shared universally within the bubble, we also find the incident edges incoming to the source, and outgoing from the sink, that are shared by all transcripts going through the bubble, to find additional shared exonic parts directly connected to the bubble.

Although, the logic has been described for two partitions, it is generally applicable for the case of pairwise $\binom{n}{2}$ comparison and multinomial comparison. For pairwise comparison, for each pair (P_1, Q_1) , D_1 reduces to $D_1 = P_1 \setminus Q_1$ and D_2 to $D_2 = Q_1 \setminus P_1$. For multinomial comparison, for each path k , D_k reduces to $D_k = P_k \setminus \bigcup_{j \neq k} P_j$, which captures exonic parts present in path k but absent from all other paths.

Reduction in complexity

The algorithm base on the containment graph described in Section X has a reduced complexity compared to exhaustively checking all the powersets of the paths which is exponential. With n nodes (paths) and m edges (subset relation) in the containment graph, the complexity of the topological sort is linear to the size of the graph $O(n+m)$, and the complexity of the recursive function is proportional to the sum of the sizes of valid lower sets. The overall complexity is $O((n+m)*L(g))$, where $L(g)$ is the number of lower-sets. For most splicing graphs with nested structures among the bubbles, we found that the total number of lower sets are significantly smaller than the size of the powerset (2^n). In practical runs, larger bubbles can still take a long time, so we allow the users to specify the limit on the size of the bubbles that are examined. The default is to only examine bubbles with less than 20 alternative paths.

The time that it takes to generate the partitions and identify the shared and distinct exonic parts is significant, but it only has to be run once for a set of annotations (gtf). We have run it for the human genome gencode ver. 34 and other frequently used genomes and annotations, and the data is shared.

We also give user the option to collapse the internal bubbles in hierarchical order. We order the bubbles so that shorter and deeper (inner nested) bubbles come first, to examine the complete set of bubbles in the splicing graphs for each gene. As longer and outer bubbles are examined, the inner and shorter bubbles nested within will be examined redundantly as part of larger bubbles many times. To avoid this redundancy, we can collapse the bubble as we traverse the bubbles from inner to outer order. After examining the inner bubble to identify the distinct and shared exonic parts, we collapse the inner bubble by retaining only the single path followed by the largest number of transcripts based on annotation, before moving to the outer bubble.

Discussion

We present a unified mathematical framework based on set operations over exonic part paths, differing primarily in their grouping strategy: pairwise treats each path independently, multinomial allows n-way comparison, and binary focuses on the meaningful two-group comparison. The choice among these approaches depends on the biological question, sample size, and computational resources available. The main limitation is that we are currently limiting the size of the bubble to less than or equal to 20. This corresponds to skipping X bubbles in the human genome. Currently the method does not handle novel splice junctions. Currently the method handles all valid paths annotated in the gtf, which we could limit it to observed paths in the future.

References

- Aquino, J. *et al.* (2025) A novel splicing graph allows a direct comparison between exon-based and splice junction-based approaches to alternative splicing detection. *Brief. Bioinform.*, **26**.
- Bindreither, D. *et al.* (2022) SplicingGraphs: Create, manipulate, visualize splicing graphs, and assign RNA-seq reads to them. *R Package Version 1380*.
- Heber, S. *et al.* (2002) Splicing graphs and EST assembly problem. *Bioinforma. Oxf. Engl.*, **18 Suppl 1**, S181-188.
- Katz, Y. *et al.* (2010) Analysis and design of RNA sequencing experiments for identifying isoform regulation. *Nat. Methods*, **7**, 1009–1015.
- Shen, S. *et al.* (2014) rMATS: Robust and flexible detection of differential alternative splicing from replicate RNA-Seq data. *Proc. Natl. Acad. Sci.*, **111**, E5593–E5601.
- Sterne-Weiler, T. *et al.* (2018) Efficient and Accurate Quantitative Profiling of Alternative Splicing Patterns of Any Complexity on a Laptop. *Mol. Cell*, **72**, 187-200.e6.
- Trincado, J.L. *et al.* (2018) SUPPA2: fast, accurate, and uncertainty-aware differential splicing analysis across multiple conditions. *Genome Biol.*, **19**, 40.
- Vaquero-Garcia, J. *et al.* (2023) RNA splicing analysis using heterogeneous and large RNA-seq datasets. *Nat. Commun.*, **14**, 1230.
- Wang, Q. and Rio, D.C. (2018) JUM is a computational method for comprehensive annotation-free analysis of alternative pre-mRNA splicing patterns. *Proc. Natl. Acad. Sci.*, **115**, E8181–E8190.
- Xing, Y. *et al.* (2004) The Multiassembly Problem: Reconstructing Multiple Transcript Isoforms From EST Fragment Mixtures. *Genome Res.*, **14**, 426–441.