Vials: Visualizing Alternative Splicing of Genes

Hendrik Strobelt, Bilal Alsallakh, Joseph Botros, Brant Peterson,
Mark Borowsky, Hanspeter Pfister, and Alexander Lex

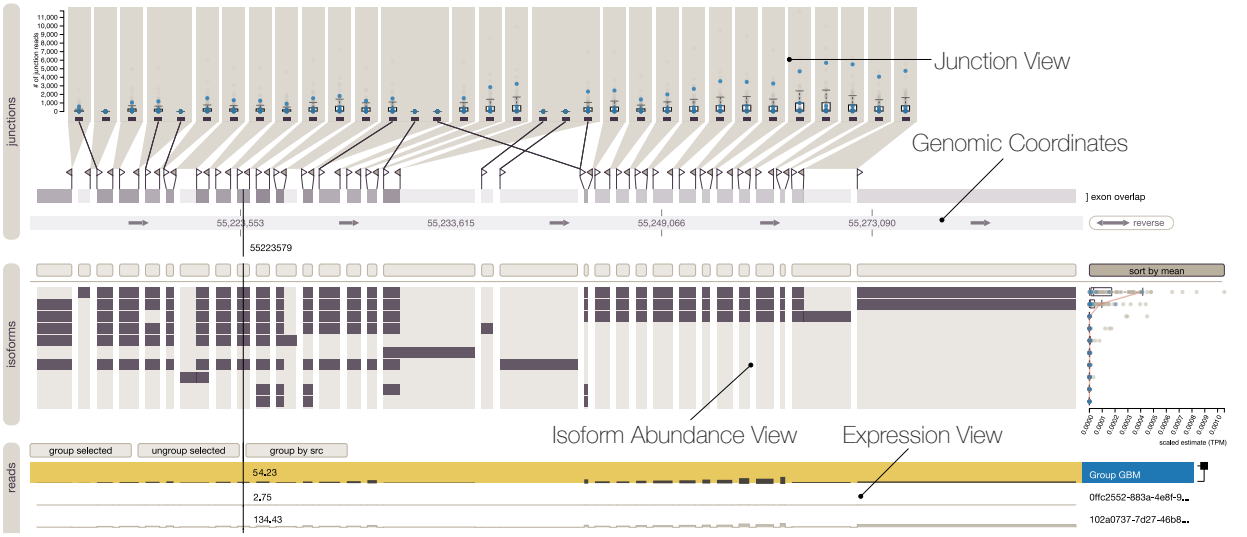


Fig. 1. Vials showing isoforms for the gene EGFR and data from The Cancer Genome Atlas.

Abstract— Alternative splicing is a process by which the same DNA sequence is used to assemble different proteins, called protein isoforms. Alternative splicing works by selectively omitting some of the coding regions (exons) typically associated with a gene. Detection of alternative splicing is difficult and uses a combination of advanced data acquisition methods and statistical inference. Knowledge about the abundance of isoforms is important for understanding both normal processes and diseases and to eventually improve treatment through targeted therapies. The data, however, is complex and current visualizations for isoforms are neither perceptually efficient nor scalable. To remedy this, we developed Vials, a novel visual analysis tool that enables analysts to explore the various datasets that scientists use to make judgments about isoforms: the abundance of reads associated with the coding regions of the gene, evidence for junctions, i.e., edges connecting the coding regions, and predictions of isoform frequencies. Vials is scalable as it allows for the simultaneous analysis of many samples in multiple groups. Our tool thus enables experts to (a) identify patterns of isoform abundance in groups of samples and (b) evaluate the quality of the data. We demonstrate the value of our tool in case studies using publicly available datasets.

Index Terms—Biology visualization, protein isoforms, mRNA-seq, directed acyclic graphs, multivariate networks

1 Introduction

Modern genome/transcriptome sequencing methods like RNA sequencing (RNA-seq) enable detailed insights into how RNA is generated from DNA. During the process of transcription from DNA to RNA certain regions (*introns*) are spliced out. The resulting mature RNA is composed of the remaining regions, i.e., the *exons*, which are also called the coding regions of a gene. This mRNA is then used as a blueprint for assembling amino acids into a protein, which in turn carries out specific functions in the cell. During this process, exons can be omitted or truncated, resulting in variations of the protein being produced, as illustrated in Figure 2. The assembly of exons into various alternative mRNA sequences is called *alternative splicing*; these alternative mRNA sequences are referred to as *isoforms*. Isoforms increase the diversity of proteins that can be produced from genes, and alternative splicing is a common biological process [21]. However, the variation in abundance of certain isoforms can be associated with diseases such as cancer [2, 8, 9].

While the process of alternative splicing has been known since the 1970s [5], large-scale and reliable detection of isoforms was elusive until recent advances in sequencing techniques. The increasing use of RNA-seq to measure *gene expression* (the abundance of specific mRNA transcripts) has also made the detection of isoforms practical. Consequently, analysis of isoform abundance will increasingly become part of the standard toolbox that researchers use to further our understanding of fundamental biological processes, the underlying nature of diseases such as cancer, and the targeted development of new and better drugs.

In this paper we introduce Vials (VIsualizing ALternative Splicing), our primary contribution, a novel visual analysis tool targeted at analyzing isoform data for large-scale RNA-seq experiments. We enable

- *Hendrik Strobelt, Joseph Botros, and Hanspeter Pfister are with Harvard University. E-mail: {hstrobelt, jbotros, pfister}@seas.harvard.edu.*
- *Bilal Alsallakh is with Vienna University of Technology. E-mail: alsallakh@cvast.tuwien.ac.at.*
- *Brant Peterson and Mark Borowsky are with Novartis Institute of BioMedical Research. E-mail: {brant.peterson, mark.borowsky}@novartis.com.*
- *Alexander Lex is with Harvard University and the University of Utah. E-mail: alex@sci.utah.edu.*

Manuscript received 31 Mar. 2015; accepted 1 Aug. 2015; date of publication 20 Aug. 2015; date of current version 25 Oct. 2015.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org.
Digital Object Identifier no. 10.1109/TVCG.2015.2467911

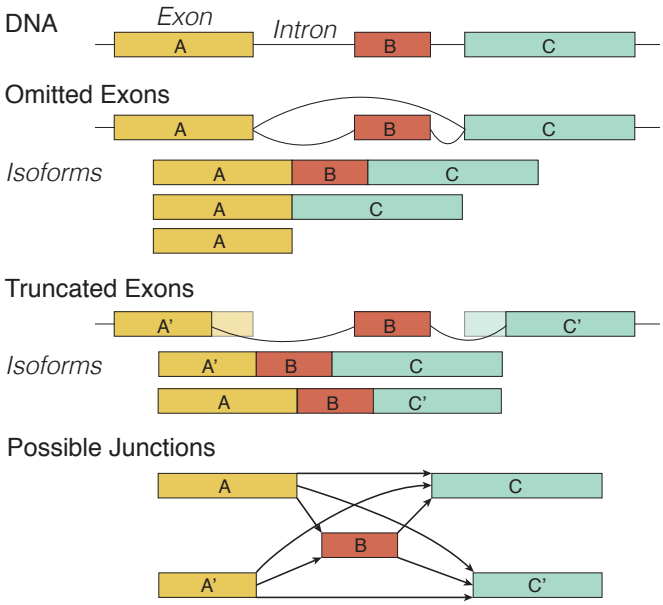


Fig. 2. The process of alternative splicing. Introns are spliced out and exons can be omitted or truncated. The final product of this process is a population of transcript isoforms. Note that only some of the possible isoforms are shown. The lower part of the figure shows all possible junctions as arrows given the three exons and their truncated variants.

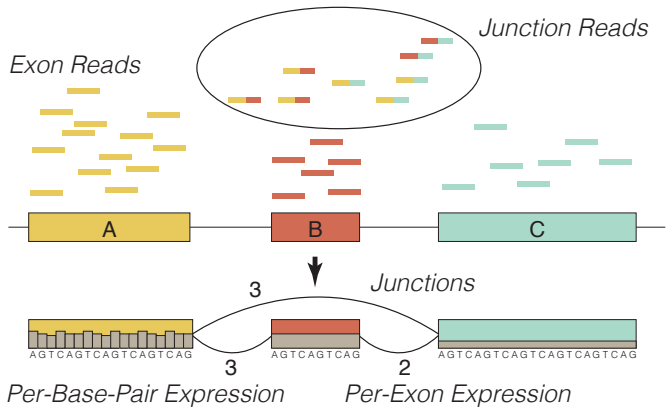


Fig. 3. Reads from a sample can either be directly mapped to an exon (exon reads) or span the junction between two exons (junction reads). By piling up the reads and counting how many reads map to a specific base-pair, expression on a per-base-pair level can be determined. By counting the reads spanning a specific junction, support for junctions can be derived. Instead of per-base-pair expression levels, some datasets provide average per-exon expression, as shown in exons B and C at the bottom of the figure.

analysts to identify isoform distributions across individual samples or groups of samples and compare them between others. In addition, we provide analysts with the means to accurately judge the quality of the data, which is critical given the early development stages of the measurement and analysis technology. Our secondary contribution is a detailed data characterization and tasks analysis for the investigation of isoforms. We translate the biological questions of our collaborators at a major pharmaceutical company into general data analysis tasks. In support of these goals, we contribute a novel method to visualize multivariate ordered graphs that emphasizes comparison between various subsets of the data.

We validate Vials using three case studies conducted by our collaboration partners, who were able to quickly confirm known findings and also discover potentially novel effects.

2 Biological Background and Data Properties

The region of DNA associated with a gene consists of an alternating succession of *exons*, the protein-coding parts of the sequence, and *introns*, the non-coding parts of the sequence. Functional gene products are created by first copying a region of DNA to produce messenger RNA (mRNA) in a process called *transcription*. This mRNA is then used as the template for assembling amino acids into functional gene products such as proteins in a process called *translation* [19].

DNA is double-stranded, i.e., there are two *strands* of complementary sequences. The two strands are identified as the *forward* and *reverse* strands. Genes are encoded on both strands of the DNA. Depictions of the genome commonly show the forward strand from left to right, which means that genes on the reverse strand have to be read from right to left. Individual genes are also occasionally depicted from left to right in reading direction, independent of the strand. Arrows are typically used to indicate the reading direction for genes.

An important part of transcription is the removal of introns and the joining of exons, a process collectively called *splicing*. Exons can be omitted or truncated, which results in a variety of different products derived from the same input sequence. It is believed that more than 80% of all genes are alternatively spliced and that this contributes significantly to the diversity of the proteome [19].

Figure 2 shows how a DNA strand can be transcribed into alternative assemblies of exons, called **isoforms**. There are eight biologically distinct types of alternative splicing events [28], yet, with the exception of retained introns, all of these types can be described in terms of omitting or truncating exons. Note that Figure 2 shows only some of the possible variations and resulting transcripts. For more details on the underlying biology we refer to the review by Matlin et al. [19].

2.1 Genomic Data Acquisition and Data Types

Isoforms can be detected using various techniques, but the most versatile and increasingly prevalent method is sequencing. Sequencing was in the past primarily used to read the DNA sequence (the genome), but advances in technology and cost reductions have made it feasible to also capture and quantify the *transcriptome* [29], i.e., the RNA. RNA-seq works by slicing the RNA in a sample into short sequences that can then be read using sequencing machines. These reads are then aligned to the genome using bioinformatics algorithms. By counting the reads that match a specific sequence its *expression level* can be determined, i.e., if many reads map to a part of the DNA, its expression is high. Figure 3 shows examples of reads that can be directly aligned to exons (in yellow, red, and green), as well as reads that span two exons, which are called junction reads. **Junctions** are the points along which exons are assembled and can be considered the "edges" between the exons, which in turn are the "nodes". In the example in Figure 3 there are three junctions (between exons A and B, B and C, and A and C). Junctions are detected when RNA fragments are found that contain the end of one exon and the beginning of another exon (the junction reads in Figure 3), suggesting that the two exons are joined in the fragment. By counting these junction reads, scientists can measure how frequently two exons are spliced together. In the example in Figure 3 there are three junction reads connecting A and B and three reads connecting A and C, but only two reads connecting B and C.

The lower part of Figure 3 illustrates expression levels on a per-base-pair level, as shown in exon A, or on a per-exon level (exons B and C). While all RNA-seq techniques provide per-base-pair expression, in many datasets this is summarized into per-exon expression values (a single expression value for the whole exon) for reasons of privacy and/or to reduce the data size.

Using this data, it is possible to infer the relative frequency of isoforms in a sample. Bioinformatics algorithms that align, quantify, and infer isoform frequencies include MISO [10], TopHat [27], and RSEM [16]. These algorithms typically provide read alignment to a reference genome, calculate expression levels, and determine the number of junction reads. In addition, they provide estimates of how often an exon is spliced into an isoform, as well as estimates for isoform abundances.

In summary, alternative splicing data is made up of three types of experimental data: (1) **isoform abundances**, capturing how much of a specific isoform there is in a sample, (2) per-exon or per-base-pair **expression data**, describing how much of a specific region of the genome is expressed, and (3) **junction support data**, representing how frequently two exons are spliced together.

In addition to these data types, there are two data sources independent from experimental data: the **reference genome**, i.e., the sequence of base-pairs that scientists agree upon as the "standard" genome for humans, and **reference information about exons and isoforms**, i.e., which exons and isoforms are known to exist. These data sources are usually taken from databases such as Ensembl[1].

2.2 Metadata

The data described in the previous section is collected and analyzed on a per-sample basis. Samples are taken from various sources, depending on the experimental design. Sometimes analysts want to compare tumor tissue with normal tissue within the same organ, while other times they are interested in different conditions, such as treatment versus no-treatment, or in investigating different phenotypes. These distinctions and other attributes (such as age, gender, etc.) are typically captured in metadata, which is available in categorical or numerical format and provides information about how the samples are related.

2.3 Data Model

The data sources used to analyze alternative splicing exhibit characteristics of common data types. The isoform abundances correspond to a table, with isoforms as rows, samples as columns, and quantitative values describing the abundances as cells.

The data describing an isoform corresponds to a binary vector, were the genetic sequence defines the order of values and the values define whether a base-pair is part of an isoform or not. In practice, the data consists of long ranges of included and excluded base-pairs. As every isoform corresponds to a binary vector, all isoforms together form a table where again the isoforms correspond to the rows.

The expression data is also based on the genetic sequence, but in contrast to the binary vectors used for isoforms, it contains scalar values. For expression data, the rows correspond to the samples, which often results in a large table. As previously mentioned, some data is aggregated on a per-exon level, which we model as all base-pairs in a region of the vector having the same values.

Finally, the junction information is best described as a multivariate, ordered, directed acyclic graph $G = (N, E)$, with nodes (N) representing all variants of exons that occur in all isoforms, and edges (E) representing junctions between exons. For each edge, we also have a vector containing the junction support for each sample. The lower part of Figure 2 illustrates such a graph. The directionality of the edges is given by the reading direction of the gene. Each isoform thus can be described as a path through the graph; some isoforms share the same nodes, while others have nodes that are not identical but cover a partially overlapping region.

3 Domain Goals and Tasks

Vials was developed in a user-centered design process over the course of ten months involving the scientific data analysis team of a major pharmaceutical company. Two of the authors of this paper are also members of that team. The development of Vials was triggered by their need to make sense of large amounts of alternative splicing data and their frustration with state of the art tools.

Based on interviews with our collaborators we identified two types of goals: finding biologically relevant insights in the data, and checking the quality and correctness of the data to establish trust. In practice, judging data quality is also an important prerequisite for exploring the data to find insights.

Hereafter we assume that the biologist has already identified a gene of interest. Interesting genes for a particular task can be found in databases or can be the output of an algorithm. Such an algorithm could, e.g., report a deviating use of isoforms for the samples under consideration. Our collaborators use an in-house bioinformatics pipeline to identify interesting candidate genes. Given this precondition we identified the following domain goals:

G1: Explore differences between samples and groups One of the biologically relevant observations our collaborators are interested in are differences between samples and groups of samples, e.g., to identify variations in isoform expression. This is interesting because it could explain an effect observed in a disease phenotype or could show the effect of differing treatments between groups. Differential expression is judged in terms of magnitude (the size of the effect) and consistency across members of a group.

G2: Discover Novel Isoforms As mentioned previously, data about exons, junctions, and isoforms is retrieved from reference databases. However, these databases do not contain all possible isoforms, as many have not yet been discovered. When analyzing data, biologists want to confirm whether the data matches the reference information, or whether there are potentially new isoform candidates.

G3: Evaluate Isoforms The biologists want to judge the impact and similarity of isoforms. When two isoforms differ by multiple exons, for example, they are more likely to have different functions than two isoforms that are identical with the exception of a short truncation.

G4: Control Data Quality The quality control (QC) goal is, as previously mentioned, an essential part of the regular exploratory process, but can also be independent from actual data analysis. QC is important to identify mistakes made by the analysis algorithms or issues with the data collection. An example for a QC process is to compare whether overall isoform abundance correlates with mRNA expression. For example, if one isoform is reported to be very common in a sample, but the exons of that isoform are not well expressed, it is likely that the reported isoform abundance value is wrong. Other QC processes include comparing the output of different algorithms (for proofreading purposes) and checking whether biological replicates behave the same way (as expected), or show deviating behavior.

3.1 Tasks

From this set of domain goals we infer two groups of tasks: those that are primarily concerned with the tabular experimental data (expression, junction support, isoform abundance; enumerated with T), and those that are concerned with the composition of isoforms (C). In the following, we describe these tasks and state the related goals.

For each of the three data types isoform abundance, exon expression, and junction support, we identify the same **tasks for the tabular experimental data (T)**.

- **T1:** Judge the magnitude of a sample or group (e.g., is the isoform highly expressed for a given sample?) [G1, G4]
- **T2:** Compare samples and identify within-group variance and outliers (e.g., is the junction support different between samples?, is the junction support within a group of samples consistent?) [G1, G4]
- **T3:** Compare groups, i.e., identify between-group variance (e.g., is an exon expressed differently between the groups?) [G1, G4]

The **tasks related to the composition of isoforms (C)** bridge the data types. The composition tasks are:

- **C1:** Identify the exons/junction that are part of an isoform. [G2, G3]
- **C2:** Identify the relationships between isoforms, e.g., find out whether they include the same or similar exons. [G2, G3]
- **C3:** Identify evidence for novel exons or isoforms that are not in the reference data. [G2]

Finally, there is the supporting task of defining sample groupings, either based on user knowledge or through data (**GR**).

As is evident from this list, comparing between groupings and exploring the connections of multiple data types are critical for this type of analysis. We have designed Vials to address these tasks so that our collaborators can answer their higher-level questions.

[1] http://www.ensembl.org/

4 Related Work

Our work is not only related to isoform and genome visualization methods, but also to techniques of multivariate network visualization, as a key part of the data is a multivariate graph. In the following we discuss these two areas of related work.

4.1 Isoform visualization

The most commonly used visualization for isoforms involving both junction support and exon reads are Sashimi plots [11], shown in Figure 4. Static sashimi plots can be generated as part of the MISO pipeline [10], an interactive variant is part of IGV [11]. Sashimi plots show both exon expression and junction support in the same plot. Exon expression is plotted as an area chart, while the junctions are represented as edges connecting the exons. The magnitude of junction support is encoded using edge weights and labels. As is evident in Figure 4, Sashimi plots suffer from multiple shortcomings. First, edge weight is not a suitable visual encoding for the wide range of data values that junction support can take. In the fourth row of Figure 4, for example, we can see two edges connecting roughly the same region, one with a labeled value of 17, the other one with a value of 346, roughly 20 times as much, but the difference in edge weight is barely perceivable. Consequently, analysts have to rely almost exclusively on labels when judging edge weights, defeating the purpose of visualization. Second, Sashimi plots do not scale to more than a few isoforms. The example in Figure 4 shows a gene with only three isoforms (indicated by the blue bars below the Sashimi plots), yet, many occlusions of edges are evident. This problem can be partially addressed by using interactive Sashimi plots, where isoforms can be selected. Third, edges connecting base-pairs in close proximity are often not visible, as there is not enough space to draw them. Fourth, comparison of junction support between samples is perceptually inefficient, as it requires the comparison of values that can only be read from labels. Finally, Sashimi plots are not well suited to visualize more than a handful of samples. We designed Vials specifically to address the shortcomings of Sashimi plots.

SpliceSeq [24] uses a similar visual metaphor but only shows data about the junctions and high-level information on whether a genomic region is coding or non-coding. SplicePlot [30] can produce Sashimi plots, and can also aggregate multiple samples into a single Sashimi plot, conveying the average expression and junction support. SplicePlot, however, does not convey information about the deviation from the average in the aggregated groups. The GTEx Portal[2] supplements a Sashimi-plot like visualization of junctions with a list of all isoforms and their abundances, resolving some of the Sashimi-plot shortcomings. However, it does not facilitate comparison between samples.

SpliceGrapher [23], a static plotting tool, separates the visualization of isoforms, reads and junctions, similar to Vials. However, SpliceGrapher is not equipped to make comparisons between multiple samples or groups of samples.

[2] http://www.gtexportal.org

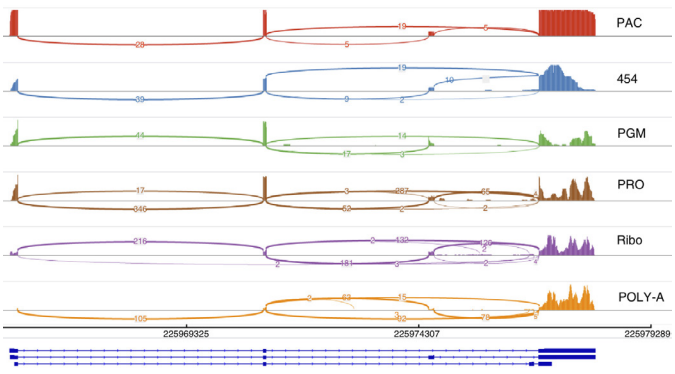


Fig. 4. Sashimi plot published in Nature illustrating differences between various analysis platforms [17]. The line width encodes edge attributes, yet the differences between small and large values are difficult to perceive. Also notice the overplotting, e.g., at the bottom right. Reprinted with permission from Macmillan Publishers Ltd.

SplicingViewer [18] visualizes splicing at a much lower level by directly showing each read aligned to the genome and indicating junctions as glyphs. SplicingViewer cannot visualize more than one sample at a time, making it unsuitable for comparison tasks.

A wide range of genome browsers and other tools show the existence and composition of isoforms as they are available in the reference databases (e.g., [7, 14]), but do not readily quantify the abundance of isoforms or junctions. A common approach for visualizing isoform abundances is to use simple plots, such as heat maps, but those simple plots fail to account for exon expression and junction data.

4.2 Multivariate Graphs

As previously mentioned, our data can be understood as a graph dataset where the exons are nodes and the junctions are edges. Rich attributes are available for both the exons and the junctions, and thus depicting this data is a multivariate graph visualization problem. We focus here on multivariate graph visualization techniques and not on the also relevant tabular data, as we use common representations for the tabular datasets, but claim a generalizable technical contribution for the multivariate graph visualization.

Multivariate graphs have received a lot of attention, as a recent state of the art survey demonstrates [12]. In this survey, biological data is identified as one of the main data sources for multivariate networks [13]. Partl et al. [22] identified four approaches to visualize multivariate networks: *on-node mapping*, i.e., directly encoding the data within the node or edge; *small multiples*, i.e., showing multiple versions of the graph that are, e.g., color-coded by the attributes; *separate linked views*, i.e., separating the topology information from the attributes into independent views, and *adapting the graph layout* to create a hybrid topology and attribute visualization. For our design, we chose to employ layout adaption to visualize the edge attributes and small multiples for the node attributes. We did not use direct encoding, as it does not scale to the desired number of samples, and avoid linked views, as they introduce a strong separation between attributes and topology.

One example for layout adaptation related to our junction view are parallel node-link bands [6]. Nodes are connected to an axis using edges, where the axis represents a dimension, similar to a parallel coordinates plot. The position of the edge-axis intersection encodes the value of the node; multiple attributes can be shown using multiple axes. Parallel node-link bands hence use position to encode the attributes of the nodes, similar to our design for edge attributes.

Another example for layout adaptation is GraphDice [1], where nodes are positioned solely based on their attributes, effectively resulting in a scatterplot, while again using position to encode the data. Pathline [20] adapts the layout of the graph by linearizing it. Next to the nodes of the linearized graph the attributes are plotted as dot plots, which is similar to our approach. Pathline, however, is limited to a single attribute per node, and uses a different linearization approach.

5 Vials Design

Figure 1 shows the three different views in Vials. The topmost is the **junction view**, which depicts the network of exons and junctions and the junction reads for all samples. Below it is the **isoform abundance view**, which shows the isoforms, the exons they include, and the predicted isoform abundance for each sample. At the bottom is the **expression view**, which displays the expression levels for all samples or for groups of samples along the genomic coordinates.

5.1 Design Principles

Our design is guided by principles partially motivated by the domain problem and partially based on visualization theory and experience.

First, we always **use the most perceptually efficient visual encoding** available for all data. We prefer, for example, position on a common scale to encode data over size or color [3], whenever possible. This differentiates Vials from other splicing visualization techniques, such as Sashimi plots, which use edge width (size) to encode junction

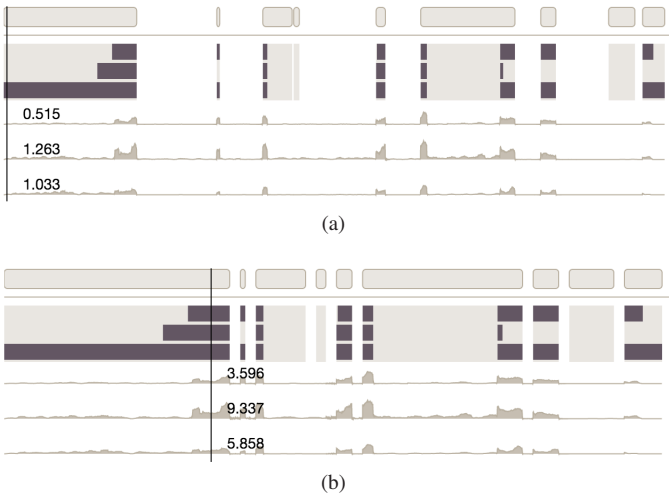


Fig. 5. Isoforms and expression of three samples on two different scales. (a) Scaled by original genome coordinates, including the introns. (b) In intron-collapsed mode (the default), where introns are reduced to a short, constant distance. The former enables a global overview of the gene, while the latter makes the exon configurations and the expression of the samples easier to read. The figures are clipped.

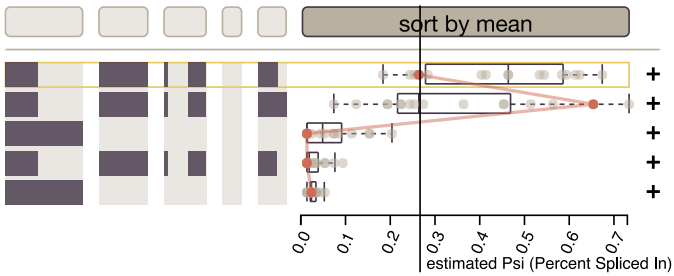


Fig. 6. The isoform abundance view shows all isoforms in a dataset as a combination of exons (left) and the abundance values of the samples for each isoform as box and dot plots on the right. The user can sort isoforms by the average abundance value (as shown in this plot), or by inclusion of an exon (isoforms that include a specified exon are moved to the top). A line connects all dots associated with a selected sample. The figure is clipped.

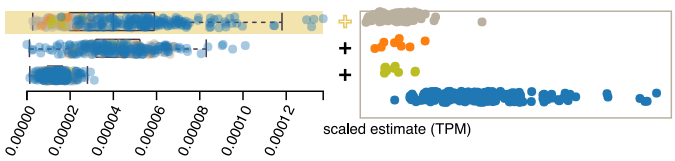


Fig. 7. Group comparison in the isoform abundance view. The samples are colored based on a grouping (orange, blue and yellow), gray samples are not grouped. For the first isoform, we show a detailed view: each group is shown in a separate row to the right of the main view.

data. Hence, our design makes use of dot plots instead of, e.g., heat maps.

Second, our design uses position to integrate information across all views by using a **shared genomic coordinate system**. This means that, for example, a specific exon is at the same horizontal position across all views. We also employ a strong coordination of views through **linking and brushing**.

Nevertheless, we decided to keep **each data type in a separate view**. This choice was partially guided by insights gained from understanding our collaborator's workflow. We observed that they investigate one data type at a time by, for example, comparing data within a data type, and only when they found something interesting do they look for supporting evidence in other data types. Separate views also allow us to address the tasks regarding the experimental data (T1-T3) in a scalable way. Showing hundreds of samples, for example, is not possible in an integrated view, like Sashimi plots.

5.2 Genomic Coordinates

The isoform abundance view, junction view, and expression view share one horizontal scale based on the genomic sequence. As can be seen in Figure 1, a reference "crosshair" line indicates the current genomic position across all views, and we periodically render genomic positions and arrows indicating the reading direction to provide orientation.

The average length of a gene is between 10k and 15k base-pairs, although the length varies significantly between genes, while exons (the coding regions) make up only a small percentage of these [25]. As the most relevant data is associated with the exons, we introduce an intron-collapse feature, shown in Figure 5. We break the genomic scale in regions where no exons occur, yet preserve the ratio of exon lengths, i.e., length comparison between two different exons remains possible. On demand, introns can be shown at full length. This design was motivated by our collaborators, who sometimes need to see the global structure of the gene, but often prefer more details for the relevant coding parts.

By default, we follow conventions and show the forward strand of the genome from left to right. However, we also allow users to reverse the reading direction. Our collaborators commented that doing so makes it easier to read genes located on the reverse strand, which would otherwise have to be read from right to left.

5.3 Isoform Abundance View

The isoform abundance view, shown in Figure 6, visualizes the binary matrix of isoforms, which is used to understand the composition and similarity of isoforms (tasks C1 and C2), and the isoform abundances used for quantification and comparison (tasks T1-T3). The exons that are part of an isoform are shown as dark blocks. Light-gray columns indicate the regions where at least one isoform expresses an exon. This allows our collaborators to quickly judge whether an exon is truncated, as can be seen in the first column of Figure 6.

Isoforms can be highlighted and selected, and these interactions are propagated to the other views. The isoforms can be interactively ranked, either by the average abundance of all samples, to see the most important isoforms on top, or by inclusion of an exon, to quickly see which isoform contains an exon of interest. The example in Figure 6 is ranked by the mean isoform abundance as indicated by the dark header bar. Sorting by exon uses a three-tier hierarchy. The first criterion is binary: isoforms containing the exon are ranked above those where it is absent. Ties are then broken by exon start position (exons that start earlier are put on top), and remaining ties are broken by coverage, i.e., larger, less truncated exons are ranked on top.

The abundance of each isoform for the samples is shown using box plots and dot plots to the right of the exons. We initially used only dot plots, however, while our collaborators were excited about the dot plots for smaller datasets, they mentioned that distributions are harder to judge for larger datasets and commented that box plots would be more intuitive for them. Hence, we added a box plot to the background of the dot plots and allowed them to switch off the dots (except for outliers) for larger datasets. To improve scalability of the dot plots we use jitter to spread the dots and transparency to reduce overplotting.

The dots can be brushed, which results in the sample being highlighted across all views. We chose to use dot plots over a heat map matrix view due to its more efficient visual encoding (position vs. color) and its superior scalability. The downside of dot plots—that patterns do not emerge across columns—can be alleviated using interactive brushing. Another alternative to the dot plots are parallel coordinates, however, given the limited available space, we found that parallel coordinates lead to significant clutter. Nevertheless, to show all values of a selected sample, a polyline is drawn across isoforms highlighting the samples value in each isoform (see Figure 6).

When groups are defined, the dots are color-coded using colors associated with the groups across all views, as shown in Figure 7. Our collaborators noted, however, that for some use cases group comparison for many dots with different colors can be difficult due to overlaps.

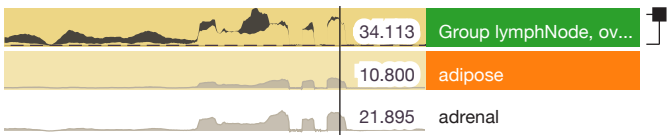


Fig. 8. Expression view showing one group on top and two individual samples in tracks. The first two tracks are selected and color-coded, which is propagated to the other views. The grouped track shows the average expression of its members plus/minus one standard deviation.

To enable accurate group comparisons using position on a common scale, the analyst can reveal a detail view: for a selected isoform, we show each group in a separate row to the right of the primary abundance plot (see Figure 7).

5.4 Expression View

The expression view, shown in Figure 8, shows the measured abundance of mRNA along the genomic coordinates for samples and groups of samples in tracks. The expression is encoded as an area chart along the genomic coordinates, a common encoding for expression in genomic data.

The expression view also serves as the main point of interaction with samples and groups. Hovering over a sample highlights this sample across other views for quick inspection. Selecting a sample or a group highlights it in color (green group and orange individual sample in Figure 8), both in the expression view and in all other views.

The expression view enables users to define groups of samples (task GR) to allow comparison between aggregated expression values. Groups can be specified manually by selecting samples and pressing the "group" button, or based on meta-data associated with each sample. The tracks belonging to a group can be collapsed, as can be seen in the first group in Figure 8. The collapsed view shows an area chart capturing the values of one standard deviation around the mean for each entry.

As previously discussed, expression data is available either on a per-base-pair or a per-exon basis. Figure 8 shows per-base-pair data, while Figure 1 shows per-exon expression. For per-base-pair data, the resolution of the data far exceeds what can be sensibly shown on the screen, hence down-sampling of the data is prudent. For details on the data processing see the supplementary material.

5.5 Junction View

The visualization of junction support in multiple samples poses the most challenges regarding efficient visualization, as discussed in Section 4. A junction connects the end of one exon and the start of a following exon. As shown in Figure 2, the junction data for one sample can be modeled as a weighted, directed, acyclic graph. An edge in this graph corresponds to a junction and its weight describes the junction support. Furthermore, the graph has a defined ordering of nodes: the exons only connect to other exons that are upstream along the genome. As we consider multiple samples, the graph is multivariate: each edge has different weights for each sample. The objective of our visualization is to compare junction support (weights) between individual samples and between groups of samples. This is interesting to our collaborators on both a global scale, e.g., to identify the most common junctions (tasks T1-T3) for certain samples, and for individual isoforms, to, e.g., judge whether the junction information corroborates the isoform predictions (task C3).

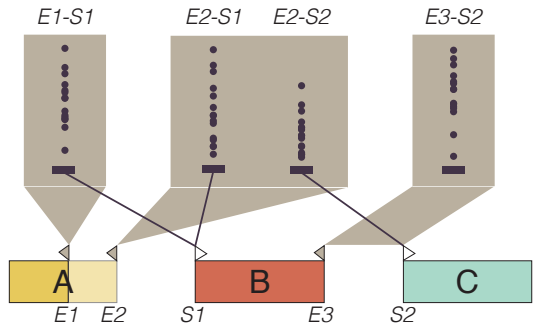
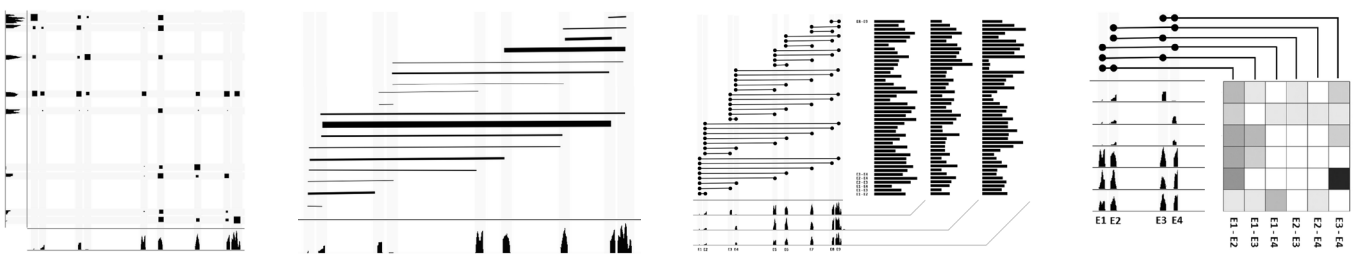


Fig. 10. Concept of the junction view. Nodes (exons) are plotted along a horizontal axis, the start (S1, S2) and end (E1-E3) of a node are indicated by triangles on top of the exons. For each end of an exon (e.g., the variants of exon A terminate at E1 and E2) we draw a polygon. Within that polygon, we display a dot-plot for each edge originating from that exon. For the exon associated with position E2, for example, there are two connections (to S1 and S2), which are indicated by explicit edges to the start of the exons. For cases where the connected exons immediately follow each other on the genomic sequence, we use a polygon connecting both exons directly (see E3-S2). The dot plots visualize the edge attributes, i.e., the support of the junctions.

We considered multiple alternative designs, including those illustrated in Figure 9 (see the supplementary material for an enlarged version of the figure). Figure 9(a) shows the support of a junction in a matrix between two samples. We abandoned this design as it is only suitable for up to two samples and is not space efficient. The design in Figure 9(b) uses lines to show edges between the exons and edge width to encode the strength of the edges. This design is similar to Sashimi plots, yet resolves some of the cluttering issues. However, like Sashimi plots, it is limited to a single sample and uses edge width, an inefficient visual encoding. Figure 9(c) uses a matrix to identify the edge and bars to encode the edge weight, in a design similar to UpSet [15]. We abandoned this design due to a lack of scalability with respect to the number of samples. Finally, Figure 9(d) shows a variation of the bar approach using a heat map instead of bars, which is more scalable than the bars but uses an inferior visual encoding (saturation vs. position).

We settled on the design illustrated in Figure 10 due to its superior scalability with respect to the number of edges and samples and its direct connection of the nodes, the links, and their attributes. Horizontally, we plot exons along the gene coordinate axis (A, B, C in Figure 10). As variants of exons can overlap, we render each exon



(a) Matrix (b) Lines (c) Bars (d) Heat Map

Fig. 9. Selected design alternatives for the junction view. The figures are reproduced in a larger format in the supplemental material. (a) Two sequences are juxtaposed and a matrix view visualizes the strengths of the connections. (b) Lines directly show the edges, where the edge weight encodes the connection strength. (c) Bars show the weight of the edges, the source and target of an edge are identified using a matrix to the left. Each bar column is associated with a single sample. (d) A heat map's rows are placed next to samples, the columns represent edges that are traced to the exons using connection lines.

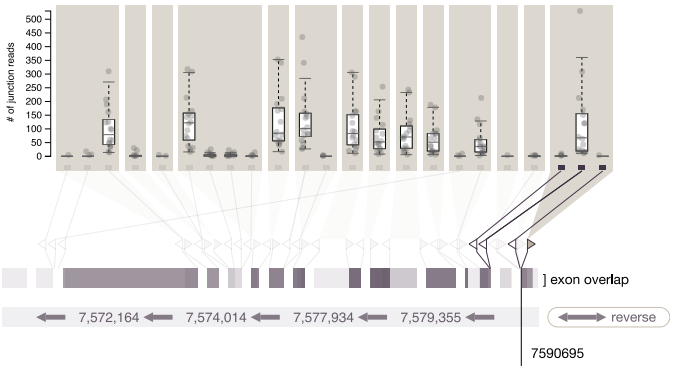


Fig. 11. An example of the junction view for the gene TP53. The exon start/end triangles are detached from the exon representation and spaced out to avoid overplotting and create visual elements suitable for interaction. Here the end of the first exon (the reading direction, indicated by the arrows, is right to left, as the gene is on the reverse strand) is selected and all unrelated junctions are faded out. We can see that this exon has edges to three other exon variants. The cursor displays the genomic location.

variant transparently, which makes the common variants stand out.

The start and end of an exon variant are represented by triangles placed at their genomic location, pointing towards their exons. The end of an exon, shown as a gray triangle, corresponds to the start of an edge (i.e., the junction); the beginning of another exon, shown as a white triangle, corresponds to the end of an edge.

We draw a box for each end of an exon variant and connect it with a polygon. As multiple edges can originate from each variant, we plot the edge weights for each edge in columns. The columns are linked with another exon's start site (see, e.g., the link from E1 to S1 in Figure 10).

To reduce clutter resulting from too many lines, we make use of a property of the data our collaborators observed: The majority of junctions in transcripts are only between adjacent exons. We simplify the visualization of these junctions by extending the polygon to bridge the gap between the adjacent exons (see E3-S2 in Figure 10).

To represent the edge weights (the junction support data) we again use dot and box plots, as we do in the isoform abundance view, making the design consistent and perceptually efficient.

Due to exon truncation, it is common that variants of exons terminate at genomic positions that are very close to each other, which would cause many of the triangles marking the start and end sites overlap. To counteract this, we detach the triangles from the exon representations and space them out so that they don't overlap. We then connect them with their original genomic location (see Figure 11), similar to the approach taken in Variant View [4]. This not only results in a better overview of exon start and end sites but also introduces visual elements (the triangles) that can be used for interaction. By hovering over a triangle, all unrelated edges are faded out, as shown in Figure 11.

Drilling Down into the Junction View While the junction view is well suited to address tasks C2 and C3 (identify relationships between isoforms, identify novel exons/isoforms), it can become dense when a dataset with many isoforms and many exons is visualized. To better support tasks T1, T2, T3 and C1 and reduce clutter we allow analysts to interactively specify a focus isoform and fade out the edges not associated with this isoform. As there is only a single edge connecting two exons when an isoform is selected, we also replace the polygon connecting the box with a direct link to the dot plot column.

Additional information in the junction view can be revealed on demand. The example in Figure 12(a) shows multiple box plots for each edge; the data is divided by a grouping of samples, which is also indicated by color, thus enabling the tasks of judging and comparing the junction support data of groups (T1-T3).

Alternatively, we can expand the dot plots showing the edge weights into a scatterplot with an animated transition. In this scatterplot, the horizontal position of the edge weights is driven by the order of the

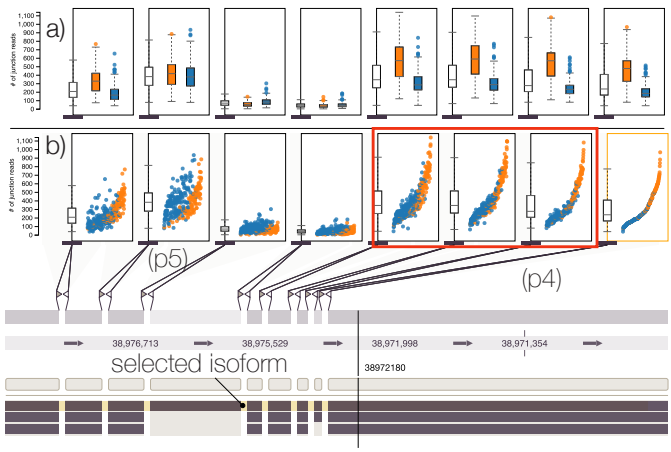


Fig. 12. Edge attribute visualization showing a novel observation in the gene SRSF7. Only the edges for the first isoform are shown. Edge attributes are colored by groups (LAML and GBM). (a) Box plots are drawn for each group. (b) The dot plots are expanded to a scatterplot and the leftmost edge is selected, which defines the horizontal position of the dots. In this edge, the dots are spaced equally and ordered by their attribute value. The order is applied to all edges, which reveals correlations to the selected edge. The edge attributes for exons where no alternative splicing occurs correlate well with the leftmost edge, yet the second edge from the right shows differential behavior. Patterns p4 and p5 are explained in a case study description (Section 6).

values of one selected edge, for example, the rightmost edge in Figure 12(b). We order the dots horizontally by their value (smallest value on the left, largest value on the right), resulting in the characteristic curve that all dots follow exactly. The horizontal position of the samples of the selected edge is propagated to all other edges, thus showing their correlation to the selected junction. In Figure 12(b), for example, we see a characteristic divide between two groups at the edge labeled p5. This feature is particularly useful to address task T2, i.e., to identify variance.

6 Case Studies

In this section we report on two case studies that demonstrate Vials' fitness for use. In addition to the two case studies described here, we also report on a third case study—how Vials can be used to understand alternative splicing in different tissues—in the supplementary material.

These case studies were chosen to demonstrate the various tasks described in Section 3. They illustrate how Vials can be used to confirm expected effects in the data and to discover novel insights. All datasets used here are available on the Vials website[3].

While feedback was elicited from multiple team members, the case studies reported on here were conducted by the domain experts that are also co-authors of this paper.

6.1 Alternative Splicing in Cancer Types

Our collaborators use publicly available data from *The Cancer Genome Atlas (TCGA)* [24] to study how cancer types differ with respect to common isoforms and exons. The TCGA project collects, publishes, and analyzes all kinds of genetic and clinical data from hundreds of patients who suffer from one of more than 20 different types of cancer.

In this case study, the experts chose to compare variants of the SRSF7 gene using samples of a brain cancer (glioblastoma, GBM, 100 samples) and a form of leukemia (acute myeloid leukemia, LAML, 167 samples), which corresponds to Goal G1—exploring differences between samples and groups. The gene SRSF7 regulates alternative splicing at a variety of targets genome-wide, while the gene itself is also regulated by alternative splicing. They chose to investigate SRSF7 because exon 4 shows large differences in how often

[3] http://vcglab.org/vials

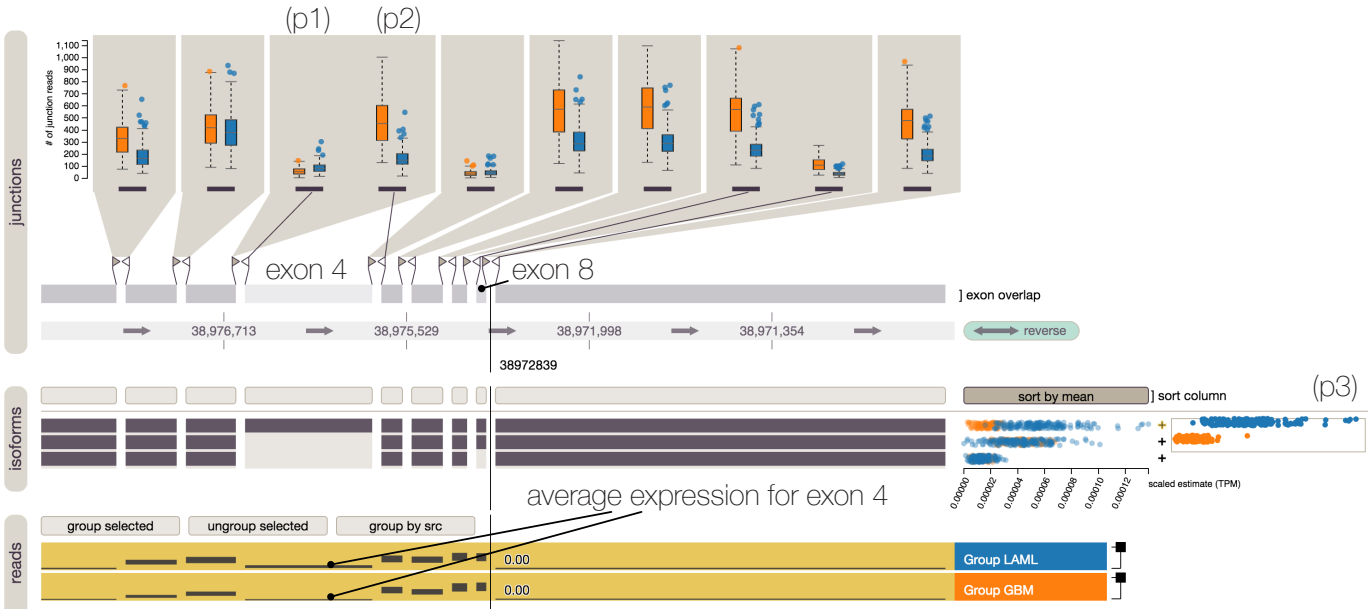


Fig. 13. Alternative splicing of the gene SRSF7. Exon 4 is alternatively spliced, i.e., only exists in the first isoform. The 100 GBM (orange) and 167 LAML (blue) samples are grouped. Notice the difference in isoform abundance: the first isoform is more common in LAML, while the second isoform is more common in GBM. The expression of exon 4 is low in both groups, yet notably lower in GBM. The junction view shows that junction support for both the edge leading to and from the alternatively spliced exon is low, but higher in LAML (blue).

it is "used" in LAML and GBM. Because differences in usage are derived from measurements of both exon expression levels and data about junctions across the two diseases, this gene is a good analysis target for the described tasks (see Section 3).

Figure 13 shows SRSF7 with GBM samples highlighted in orange and LAML samples highlighted in blue. The TCGA data provides expression data as an average for every exon, as is evident from the constant blocks in the expression view in Figure 13. The expression data in Figure 13 is aggregated into the two disease groups LAML and GBM. When exploring this data, the domain expert noted that there is roughly equivalent expression of the exons that are not alternatively spliced between the two groups (e.g., exons 3,5,6,7). In contrast, the alternatively spliced exon 4 shows very low expression in GBM, but some expression in LAML (task T3 applied to the expression data). Consistent with this, our collaborators observed in the junction view (top), that there is greater support for the junction joining exon 3 to exon 4 in LAML (blue) than GBM (orange) (Fig. 13, pattern p1; task T3 applied to junction support). On the other hand, GBM samples show more support for the splice junction that skips exon 4 (higher orange values in pattern p2). This confirms that both exon abundance and junction use support a difference in exon 4 splicing between LAML and GBM. Additional evidence is visible in the isoform abundance view for the first isoform (pattern p3). This isoform is characterized by the inclusion of exon 4 (task T3 applied to isoform abundance, combined with tasks C1 and C2). As expected from exon and junction data, this isoform is more abundant in LAML samples than in GBM (the blue dots show larger values than the orange dots in p3).

Potential Novel Isoform In addition to these differences, exon 8 of SRSF7 in the TCGA data is known to have weak but statistically significant alternative splicing [24]. While the support for differences in expression of this exon between GBM and LAML is small, exploring the junctions associated with exon 8 lead our collaborators to a new hypothesis regarding a yet unknown exon variant. Specifically, in ranking samples by support for the exon 8 – exon 9 junction (p4 in Figure 12), they observed that levels of non-alternatively-spliced junctions are generally highly correlated with this junction in both cancer types (both orange and blue samples show approximately equivalent correlation in the boxed scatterplots in Figure 12(b)). In contrast, two visually distinct populations emerge in the exon 2 – exon 3 junction (p5). Specifically, LAML samples (blue) show a greater and apparently linear relationship with the exon 8 – exon 9 junction, while GBM samples (orange) display lower exon 2 – exon 3 junction use proportional to the exon 8 – exon 9 junction in the starred scatterplot. One potential explanation for this observation is an alternative transcription start site which is absent from the gene reference database, and which does not use exons 1 or 2, leading to an observation of the type of Goal G2—discover novel isoforms. This hypothetical alternative starting exon would be connected with a junction to exon 3.

6.2 Quality Control

While exploring the data, our collaborators were continuously looking for issues of data quality (goal G4). They eventually found a striking case of missing and wrong data in the gene EGFR in the Bodymap[4] dataset. Figure 14 shows a case where the white blood cell sample, highlighted in red in Figure 14(a), shows strongly deviating behavior from other samples in the isoform abundance view. Initially intrigued, our collaborators quickly identified that this is a data quality issue, as there is no expression data available for the white blood cell sample, as is evident when inspecting the data in the expression view, shown in Figure 14(b). Similarly, there is no junction support for this sample, indicating that the reported isoform abundances are an artifact of the processing pipeline. While this is an extreme case of a data quality problem, it would not be immediately apparent when only the isoform abundances are investigated.

7 Discussion

Our collaborators noted that Vials is an ideal companion to the many tools that compute abundances of exons and isoforms, as it allows them to "drill down" from the isoform-level differences into the specific junction data that determine the identity and abundance of each transcript isoform. They were excited that Vials provides a mechanism for both visualizing abundance of each isoform and comparing isoforms at the level of exon abundance and junction use, directly from primary data. They specifically highlighted the importance of analyzing many and grouped samples, a feature competing methods lack and which they identified as their primary limitation.

[4] http://www.ebi.ac.uk/arrayexpress/experiments/E-MTAB-513/

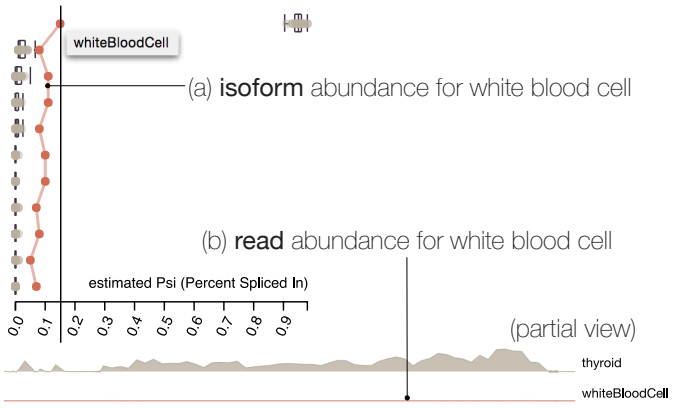


Fig. 14. Example for control of data quality in EGFR using Bodymap. The white blood cell sample can be easily spotted as outlier with respect to isoform abundance and read abundance.

Regarding learnability, both, the group of people more closely involved with the development, but also others on the team commented that they found the tool generally straight-forward to use, with the junction view being harder to understand than the other views of Vials. However, after they became familiar with the representation, they reported to be very satisfied with its expressiveness.

While we believe that Vials is an efficient and useful alternative splicing visualization technique it has some limitations. Compared to Sashimi plot-like approaches, where all the information related to one sample is in a single place, it is more difficult in Vials to see all the characteristics of a single sample, as they are distributed over multiple views. However, Sashimi plots and Vials have different use cases: Vials is designed as an exploratory data analysis tool for the visualization of many samples, whereas Sashimi plots are better suited for visualization of individual samples.

Lessons Learned The iterations on this project taught us some interesting lessons about working with genomic data. First, we discovered that most data we dealt with is complex, i.e., it does not fit easily into a simple conceptual model. For example, early in the project we expected an isoform to be a well-defined combination of a set of exons, and we expected isoforms to differ only in terms of which of these exons they use. In reality, however, exon start and end-sites can vary by "a couple of base-pairs", which meant that we had to deal with many more variants of exons than we initially expected. This observation triggered the development of the flags indicating start and end sites of exons.

Another insight is that there are many competing data formats for representing biological data, which necessitates a highly flexible preprocessing pipeline. These different data formats sometimes even require that the visual representations are flexible. For example, the expression view in Vials can display per-base-pair reads and per-exon reads using the same visual encoding.

Generalization While the topic of visualizing alternative splicing is very important, it is also highly specialized, and consequently the combination of views used in Vials is also specialized. However, we believe that individual components of Vials translate well to other domains and datasets. The most interesting example is the junction view, which visualizes a highly multivariate graph. Similar data characteristics can be found, for example, in time-series data, where potentially overlapping events correspond to the nodes (exons) in the graph, and information about these events (e.g., attributes of attendees) could be visualized using the dot plots, or other visual representations.

Scalability Vials scales to the vast majority of genes, as most genes have fewer than 10 exons and the vast majority of genes have 20 exons or less [25]. Yet there are outliers—the largest known human gene (TTN) has more than 300 exons. We consider about 30 junctions to be the limit up to which Vials can be used without restrictions. For more than 30 junctions, scrolling becomes necessary.

With respect to the number of samples, Vials is very scalable. We demonstrate a case study with 276 samples which is considered a large study of mRNA-seq data by current standards, which results in no complications for the isoform or junction view. To handle larger numbers of samples, we will not plot individual data points, but instead use the box plots as aggregates. For more than about 20 samples, the expression view can require a lot of scrolling. This, however, can easily be remedied by grouping the samples.

8 Implementation

Vials is a web-based, open source visualization with a D3/JavaScript front-end and a Python back-end that enables sophisticated computation and data management. Our prototype is based on the Caleydo Web framework[5]. The Python server component runs in a Vagrant virtual box and is therefore separated from the rest of the machine it runs on, thus easing deployment on different platforms. On the client side the framework uses require.js to modularize its components. An event handling mechanism enables sophisticated view coordination.

A client-only prototype that includes the datasets used for the figures and case studies in this paper, and links to the source code of vials are available at `http://vcglab.org/vials/`.

9 Conclusion and Future Work

Vials is the first visualization technique that allows analysts to explore isoforms and their properties for a large number of samples and to flexibly compare groups of samples. It is also distinct from prior work as it emphasizes the use of perceptually efficient visual encodings. Vials integrates all information needed for isoform analysis including isoform abundance data, per-exon/per-base-pair expression data, and junction support data. It provides an overview of hundreds of samples but also enables experts to drill down into the data associated with individual samples.

We have shown in two case studies that Vials is fit for use in real-world scenarios. For a more efficient workflow of our collaborators, we plan to integrate Vials with their current tools, which will allow them to easily and interactively explore the isoforms of genes or exons reported as interesting by their bioinformatics pipelines.

While Vials as a whole is a specialized technique, we believe that its individual views, principles, and the lessons we have learned translate to other datasets and domains. For example, we plan to generalize the junction view to work with similar multivariate graph datasets.

In the future, we plan on integrating Vials with additional data types and visualizations. For example, it would be desirable to see how mRNA-seq data is correlated with pathways, mutation data, and/or copy number variation data. Similarly, it would be valuable to see clinical data, such as survival plots, in the context of groupings derived from isoform abundances [26].

Acknowledgments

We thank Samuel Gratzl for help with the framework, Nils Gehlenborg for comments on the manuscript, and Sebastian Hörsch and other members of the SDA team for their expertise and feedback. This work was supported in part by Novartis Institutes for BioMedical Research, the Austrian Science Fund (J 3437-N15), the Air Force Research Laboratory and DARPA grant FA8750-12-C-0300, and the US National Institutes of Health (U01 CA198935)

References

[1] A. Bezerianos, F. Chevalier, P. Dragicevic, N. Elmqvist, and J. D. Fekete. GraphDice: A System for Exploring Multivariate Social Networks. *Computer Graphics Forum (EuroVis '10)*, 29(3):863–872, 2010.

[2] L. H. Boise, M. González-García, C. E. Postema, L. Ding, T. Lindsten, L. A. Turka, X. Mao, G. Nuñez, and C. B. Thompson. bcl-x, a bcl-2-related gene that functions as a dominant regulator of apoptotic cell death. *Cell*, 74(4):597–608, 1993.

[5] `http://caleydo.org`

[3] W. S. Cleveland and R. McGill. Graphical Perception: Theory, Experimentation, and Application to the Development of Graphical Methods. *Journal of the American Statistical Association*, 79(387):531–554, 1984.

[4] J. Ferstay, C. Nielsen, and T. Munzner. Variant View: Visualizing Sequence Variants in their Gene Context. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '13)*, 19(12):2546–2555, 2013.

[5] M. B. Gerstein, C. Bruce, J. S. Rozowsky, D. Zheng, J. Du, and et al. What is a gene, post-ENCODE? History and updated definition. *Genome Research*, 17(6):669 –681, 2007.

[6] S. Ghani, B. C. Kwon, S. Lee, J. S. Yi, and N. Elmqvist. Visual Analytics for Multimodal Social Network Analysis: A Design Study with Social Scientists. *IEEE Transactions on Visualization and Computer Graphics (VAST '13)*, 19(12):2032–2041, 2013.

[7] E. D. Harrington and P. Bork. Sircah: a tool for the detection and visualization of alternative transcripts. *Bioinformatics*, 24(17):1959–1960, 2008.

[8] S. Hoersch and M. A. Andrade-Navarro. Periostin shows increased evolutionary plasticity in its alternatively spliced region. *BMC Evolutionary Biology*, 10(1):30, 2010.

[9] J. N. Honeyman, E. P. Simon, N. Robine, R. Chiaroni-Clarke, D. G. Darcy, I. I. P. Lim, C. E. Gleason, J. M. Murphy, B. R. Rosenberg, L. Teegan, C. N. Takacs, S. Botero, R. Belote, S. Germer, A.-K. Emde, V. Vacic, U. Bhanot, M. P. LaQuaglia, and S. M. Simon. Detection of a Recurrent DNAJB1-PRKACA Chimeric Transcript in Fibrolamellar Hepatocellular Carcinoma. *Science*, 343(6174):1010–1014, 2014.

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

[11] Y. Katz, E. T. Wang, J. Silterra, S. Schwartz, B. Wong, J. P. Mesirov, E. M. Airoldi, and C. B. Burge. Sashimi plots: Quantitative visualization of RNA sequencing read alignments. *arXiv:1306.3466 [q-bio]*, 2013.

[12] A. Kerren, H. C. Purchase, and M. Ward, editors. *Multivariate Network Visualization*. Number 8380 in Lecture notes in computer science. Springer, 2014.

[13] O. Kohlbacher, F. Schreiber, and M. O. Ward. Multivariate Networks in the Life Sciences. In *Multivariate Network Visualization*, number 8380 in Lecture Notes in Computer Science, pages 61–73. Springer, 2014.

[14] R. M. Kuhn, D. Haussler, and W. J. Kent. The UCSC genome browser and associated tools. *Briefings in Bioinformatics*, 14(2):144–161, 2013.

[15] A. Lex, N. Gehlenborg, H. Strobelt, R. Vuillemot, and H. Pfister. UpSet: Visualization of Intersecting Sets. *IEEE Transactions on Visualization and Computer Graphics (InfoVis '14)*, 20(12):1983–1992, 2014.

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

[17] S. Li, S. W. Tighe, C. M. Nicolet, D. Grove, S. Levy, W. Farmerie, A. Viale, C. Wright, P. A. Schweitzer, Y. Gao, D. Kim, J. Boland, B. Hicks, R. Kim, S. Chhangawala, N. Jafari, N. Raghavachari, J. Gandara, N. Garcia-Reyero, C. Hendrickson, D. Roberson, J. A. Rosenfeld, T. Smith, J. G. Underwood, M. Wang, P. Zumbo, D. A. Baldwin, G. S. Grills, and C. E. Mason. Multi-platform assessment of transcriptome profiling using RNA-seq in the ABRF next-generation sequencing study. *Nature Biotechnology*, 32(9):915–925, 2014.

[18] Q. Liu, C. Chen, E. Shen, F. Zhao, Z. Sun, and J. Wu. Detection, annotation and visualization of alternative splicing from RNA-Seq data with SplicingViewer. *Genomics*, 99(3):178–182, 2012.

[19] A. J. Matlin, F. Clark, and C. W. J. Smith. Understanding alternative splicing: towards a cellular code. *Nature Reviews Molecular Cell Biology*, 6(5):386–398, 2005.

[20] M. Meyer, B. Wong, M. Styczynski, T. Munzner, and H. Pfister. Pathline: A Tool For Comparative Functional Genomics. *Computer Graphics Forum (EuroVis '10)*, 29(3):1043–1052, 2010.

[21] Q. Pan, O. Shai, L. J. Lee, B. J. Frey, and B. J. Blencowe. Deep surveying of alternative splicing complexity in the human transcriptome by high-throughput sequencing. *Nature Genetics*, 40(12):1413–1415, 2008.

[22] C. Partl, A. Lex, M. Streit, D. Kalkofen, K. Kashofer, and D. Schmalstieg. enRoute: Dynamic Path Extraction from Biological Pathway Maps for Exploring Heterogeneous Experimental Datasets. *BMC Bioinformatics*, 14(Suppl 19):S3, 2013.

[23] M. F. Rogers, J. Thomas, A. S. Reddy, and A. Ben-Hur. SpliceGrapher: detecting patterns of alternative splicing from RNA-Seq data in the context of gene models and EST data. *Genome Biology*, 13(1):R4, 2012.

[24] M. C. Ryan, J. Cleland, R. Kim, W. C. Wong, and J. N. Weinstein. SpliceSeq: a resource for analysis and visualization of RNA-Seq data on alternative splicing and its functional impacts. *Bioinformatics*, 28(18):2385–2387, 2012.

[25] M. K. Sakharkar, V. T. Chow, and P. Kangueane. Distributions of Exons and Introns in the Human Genome. *In Silico Biology*, 4(4):387–393, 2004.

[26] M. Streit, A. Lex, S. Gratzl, C. Partl, D. Schmalstieg, H. Pfister, P. J. Park, and N. Gehlenborg. Guided visual exploration of genomic stratifications in cancer. *Nature Methods*, 11(9):884–885, 2014.

[27] C. Trapnell, L. Pachter, and S. L. Salzberg. TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, 25(9):1105–1111, 2009.

[28] E. T. Wang, R. Sandberg, S. Luo, I. Khrebtukova, L. Zhang, C. Mayr, S. F. Kingsmore, G. P. Schroth, and C. B. Burge. Alternative isoform regulation in human tissue transcriptomes. *Nature*, 456(7221):470–476, 2008.

[29] Z. Wang, M. Gerstein, and M. Snyder. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10(1):57–63, 2009.

[30] E. Wu, T. Nance, and S. B. Montgomery. SplicePlot: a utility for visualizing splicing quantitative trait loci. *Bioinformatics*, 30(7):1025–1026, 2014.