LEHRSTUHL FÜR EXPERIMENTELLE BIOINFORMATIK

TECHNISCHE UNIVERSITÄT MÜNCHEN

Final Report of the Fortgeschrittenen Praktikum

Evaluation of differential Splicing tools

Alexander Dietrich

Dr. Markus List, Dr. Olga Tsoy, Prof. Dr. Jan Baumbach

Contents

1 Introduction

1.1 Alternative Splicing

Splicing describes one possible post-transcriptional modification of eukaryotic mRNA. The results of Alternative Splicing are different isoforms of the mRNA for a single gene, an isoform being a distinct selection of coding sequence regions, the exons. These different isoforms lead to different proteins with potentially different functions encoded by the same gene (figure 1.1). Alternative Splicing can appear in many different types and combinations of events, which will be explained in detail in the next chapter.
As a result the number of different proteins in a mammal can exceed the number of genes by a factor of four [1], which shows the importance of alternative splicing regarding protein diversity. Also many diseases are presumed to be linked to irregularities in this complex process, e.g. muscle dystrophy [2]. All the different mechanisms, which influence alternative splicing regulation, are still far from being fully understood [3]. Alternative splicing detection tools aim to identify splicing events, most commonly by using RNA-seq data.

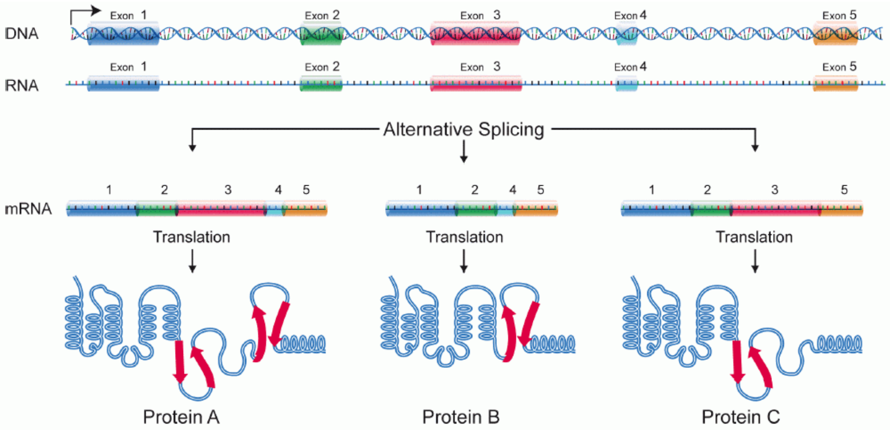

Figure 1.1: Alternative splicing enables a single gene to encode multiple proteins [4].

1.1.1 Alternative Splicing Event types

There are seven types of alternative splicing in human: exon skipping (ES), intron retention (IR), alternative 5′/donor splice site (A5), alternative 3′/aceptor splice site (A3), mutually exclusive exons (MEE), multiple exon skipping (MES), alternative first exon (AFE) and alternative last exon (ALE) (figure 1.2).
Exon skipping describes the event, when a single exons surrounded by 2 other exons is removed by splicing, resulting in two separate exons being joined together; multiple exon skipping is the same principle, only more than one exon is removed. Intron retention occurs, when a single intron is not spliced out. Alternative 3′ and 5′ splice site describes the event, in which the position of the splice site at the 3′ or 5′ end of an exon is changed. When two ES events are not independent anymore, but rather are executed in coordination (one ES event is only performed, if the other event is not performed), this is called mutually exclusive exons. Alternative last and first exon simply is the case, when the first or last exon of a gene is spliced in or not.

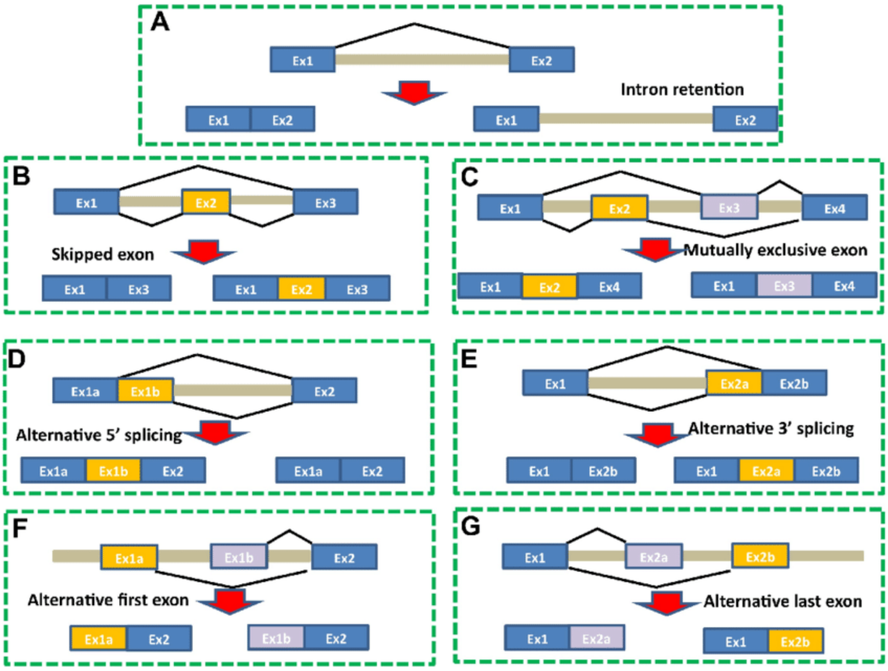

Figure 1.2: Schematic of different alternative splicing event types [5]

1.2 Bioinformatic Tools to detect Splicing Events

Many bioinformatic tools exist, which have to goal to detect these different splicing events. The main principle usually is to use an existing genome annotation of an organism and optionally use RNA-seq data (short reads) to augment this annotation. With this process novel and annotated splicing events can be located and categorized. First the annotation is used to build a so called splice-graph (figure 1.3), which is a graph representation of all different transcripts a gene can form.
Table 1.1 shows a list of different tools and their basic principle on how to detect splicing events.

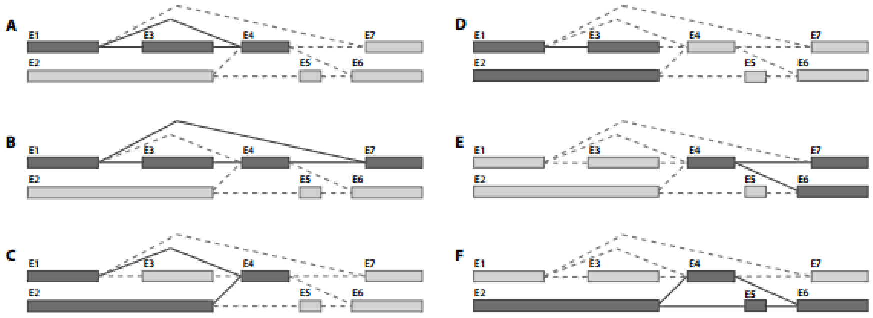

Figure 1.3: The splice graph representation of different splicing events. Boxes represent exons, solid or dashed lines the possible intron variants. The dark boxes and solid lines show the event of interest: A: ES, B: MES, C: A5, D: IR, E: A3, F: MEE [6]

1.3 Motivation

Initially each of the above mentioned tools aims to do the same overall thing: finding alternative splicing events; but each tool has its own approach and therefore its own way of generating output files for the found events and annotating an event. This makes it really hard for a user to compare multiple tools. The goal of project was to create an easy to use tool, which can handle those different tool outputs and create a single new file for each, without loosing information. The new file will then have the same structure for all alternative splicing event detection tools, making precise comparisons per gene or on coordinate level possible.

name	principle/strategy	latest release	language
ASGAL[7]	Splicing graph	2020	C++, python
ASpli [8]	Gene Features	2017	R
CASH [9]	Gene Features	2015	java
EventPointer [10]	Splicing Graph	2020	R
IRFinder [11]	Gene Features	2020	C++
KisSplice [12]	Splicing Graph	2020	C++
MAJIQ [13]	Splicing Graph	2019	python
SGSeq [14]	Splicing Graph	2020	R
spladder [6]	Splicing Graph	2019	python
Whippet [15]	Splicing Graph	2019	julia

Table 1.1: Alternative Splicing event detection tools

2 Methods

During this project, four alternative splicing event detection tools were used to transform their output into a unified version, so that each tools output essentially looks the same in the end. These four tools are MAJIQ, SplAdder, ASGAL and Whippet.

2.1 Unifying outputs of Alternative Splicing event detection tools into a single format

In order to achieve easy comparison between tools, one file-format was decided on. A tab separated file will store every event, one tool finds, each event encoded by one of the seven explained standard event types. Additionally, for each event the gene name, chromosome, strand as well as a unique ID (usually the ID the tool already gives, with some minor additions in some cases) will be listed.

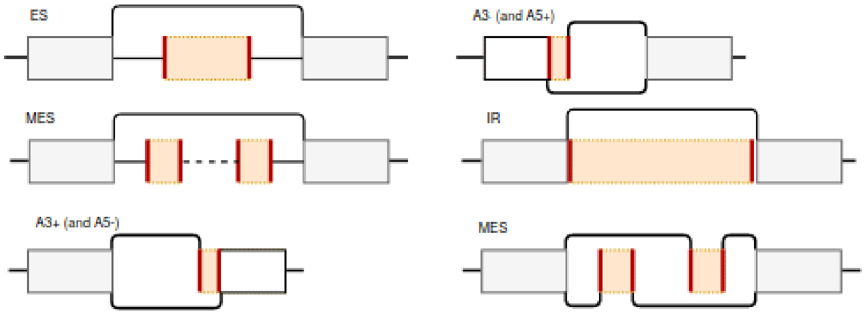

Figure 2.1: Shown are six event types and how they are annotated in the new file format. The orange boxes show the exon (or part of exon), on which the event is occurring, grey boxes are its neighbors. The red lines are the coordinates, which are stored in the file. Note that for A3 and A5 events, the strand has to be considered as well (see supplementary figure 5.1).
A detailed text version of each event can be found in the supplementary table 5.1.

The most important part of the file are the genome coordinates for each event. The following figure 2.1 will explain in detail, which coordinates are reported for each event type. Event types with more than one start and stop coordinate (MES and MEE), a comma separated list of start or stop coordinates will be given.
In order to create such a format, a python tool was implemented, called OUTPUT_TRANSFORMER. The next sections will each explain how it re-calculates the output of each tool into the proposed unified file format. The exact specifications for each column can be found in the supplementary table 3.1.
There are two run modes for the OUTPUT_TRANSFORMER: `create` and `compare`. The first one can be used to transform the output of one or more tools into the unified version, for each tool a separate command line flag is used to give the path to the tools output file or directory. The second mode will be explained in section 2.3.

2.1.1 MAJIQ `-m`

MAJIQ [16] is a software package in the python programming language to detect splicing events. They use a more flexible definition of these events in the form of "local splicing variations" (LSVs), which describes exons, from which (or to which) splicing events start (or end). This means in their definition, a LSV can contain several combinations of splicing events. A binary LSV for example would then simply be a single exon skipping event (an example visualization of such a case can be seen in supplementary figure 5.2). More complex LSVs can contain multiple "standard" event types, like one ES and one A5 event as seen in figure (2.2).

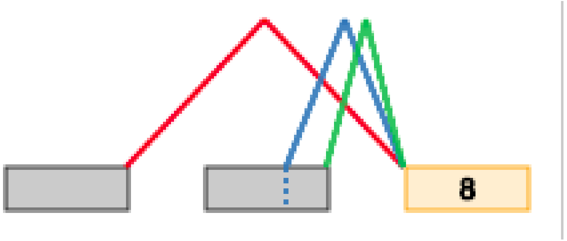

Figure 2.2: MAJIQ LSV of an ES (red) and A5+ (blue) event

The MAJIQ project also includes the VOILA tool, which is a way of generating visualizations of splicing variations as well as providing a more detailed output format than MAJIQ does: The used MAJIQ output file is called `*.psi.tsv`. Since the exon coordinates are also important for the correct event annotation (figure 5.2), but this file does not store them, the output of VOILA (created by `voila tsv` command) has to be used as well. The output of both tools can be seen in supplementary table 5.2. So the first step of the OUTPUT_TRANSFORMER is to merge the event of these two files by the event id. The resulting `merged_majiq.tsv` file is described in 2.1. With it the following events can be calculated:

column-name	explanation	origin file
gene_id	gene name	VOILA
lsv_id	unique ID for LSV	VOILA
num_junctions	Number of junctions in LSV	VOILA
num_exons	Number of exons in LSV	VOILA
de_novo_junctions	number of junctions found de novo	VOILA
seqid	label of chromosome	VOILA
strand	+ or - for strandness of LSV	VOILA
junctions_coords	list of junction coordinates; each list element has with start and stop coordinate	VOILA
exons_coords	list of exon coordinates;	VOILA
ir_coords	start and stop coordinates of IR event(s)	VOILA
A5SS	`TRUE or FALSE`: if LSV contains one or more A5 events	MAJIQ
A3SS	`TRUE or FALSE`: if LSV contains one or more A3 events	MAJIQ
ES	`TRUE or FALSE`: if LSV contains one or more ES events	MAJIQ

Table 2.1: column explanation for `merged_majiq.tsv` file with the origin of each column; important columns for event calculation are **highlighted**

IR

Use same coordinates as in the `ir-coords` column

ES & MES

If column `ES` is `TRUE`, then check for each junction in `junction_coords` if this junction is starting before one exon and ending after this exon (exon from column `exons_coords`). If this junction spans more than one exon in that way, a MES event is found.
In some cases MAJIQ reports the same ES event, but only starting from different directions. Meaning once it gets detected by the source exon in front of the skipped exon and once it gets detected by the target exon behind it. The final output file should not have duplicate events, so a unique set of events (uniqueness is a combination of gene, start, end and event type) per gene was created while reading the MAJIQ file; that way no duplicate events are stored.

A3

If column A3SS is TRUE, then check for each junction in junction_coords if it starts or ends (depending on strandness) inside of an exon from exons_coords. Then a A3 event is found.

A5

If column A5SS is TRUE, then check for each junction in junction_coords if it starts or ends (depending on strandness) inside of an exon from exons_coords. Then a A5 event is found.

Since these are representations of LSVs, all of the above cases can be present in the same line and have to be checked. It is also possible that on event type appears multiple times in the same line. MAJIQ sometimes reports "nan" coordinates, these were just adopted.

2.1.2 SplAdder -s

SplAdder [6] is a python package, which analysis alternative splicing based on RNA-seq data. They stick the standard type annotation of splicing events as described in 1.1.1, but do not report any AFE or ALE events. SplAdder creates one file per event type. The important columns of these files are can be seen in table 2.2. With these columns, all five remaining standard event types can be calculated:

IR

Use the intron_start and intron_end column values.

ES & MES

Use the exon(s)_start and exon(s)_end column values.

MEE

For the first exon: exon_alt1_start and exon_alt1_end; for the second exon: exon_alt2_start and exon_alt2_end.

column-name	explanation
contig	label of chromosome
strand	+ or - for strandness of event
event_id	unique ID for event
gene_name	gene name
columns for MES and ES (and IR)	
exon_pre_start	start of previous exon
exon_pre_end	end of previous exon
exon(s)_start (intron_start)	start of exon, where event is located
exon(s)_end (intron_start)	end of exon, where event is located
exon_aft_start	start of next exon
exon_aft_end	end of next exon
[...]	columns with coverage values of event
columns for A3 and A5	
exon_const_start	start of exon with no alternative part
exon_const_end	end of exon with no alternative part
exon_alt1_start	first possible start of alternative exon
exon_alt1_end	first possible end of alternative exon
exon_alt2_start	second possible start of alternative exon
exon_alt2_end	second possible end of alternative exon
columns for MEE	
exon_pre_start	start of previous exon
exon_pre_end	end of previous exon
exon1_start	start of first mutually exclusive exon
exon1_end	end of first mutually exclusive exon
exon2_start	start of second mutually exclusive exon
exon2_end	end of second mutually exclusive exon
exon_aft_start	start of next exon
exon_aft_end	end of next exon

Table 2.2: The different column names for SplAdders output files; each type has its own output file.

A3

If on positive strand: start is `exon_alt2_start`, end is `exon_alt1_start`; for negative strand: `exon_alt1_end` as start and `exon_alt2_end` as end.

A5

If on positive strand: start is `exon_alt1_end`, end is `exon_alt2_end`; for negative strand: `exon_alt2_start` as start and `exon_alt1_start` as end. (inverted to A3 events)

2.1.3 ASGAL `-a`

The "Alternative Splicing Graph ALigner" - ASGAL [7] uses a splice-graph and mapped RNA-seq reads to detect splicing events. An example of its output can be seen in 2.3. The information in this output file is not enough though to calculate each event-type, so to get all events correctly, the gtf annotation file was used as additional help. Even with that additional file, it was only possible to calculate IR, ES, MES, A3 and A5 events.

column-name	explanation
Type	event type
Start	start coordinate of spanning junction
End	end coordinate of spanning junction
Support	number of reads mapped to this event
Transcripts	name of transcripts with this event
file	path to different file with same information about this event

Table 2.3: Columns of the ASGAL output file. The coordinates in this case correspond to the start and end of a alternative junction (this would be the "larger" junction in each of the examples from figure 5.1)

IR

Could just adopt the `Start` and `End` columns as start and stop.

ES & and MES

Find all exons in the gtf, whose start and stop coordinate lie within the values of the `Start` and `End` columns; if more than one exon is found to fit in, a MES event is found. Report start and end of the exons from gtf.

A3+ and A5-

While iterating over gtf exons check: If `Start`-1 is equal to the end of the current gtf exon there are two options: 1) the start of the next exon is smaller than the `End`-column value; then a A3+ or A5- event is created from the start of the next exon until the

End-column value. 2) If this is not the case, then the event is created the other way around.
It can be the case, that both of these values are the same, then the end-coordinate of the event is set to "nan".

A3- and A5+

While iterating over gtf exons check: If End+1 is equal to the start of the next gtf exon there are two options: 1) the Start-column value is smaller than the end of the current exon; then a A3- or A5+ event is created from the Start-column value until the end of the current exon. 2) If this is not the case, then the event is created the other way around.
It can be the case, that both of these values are the same, then the start-coordinate of the event is set to "nan".

Since ASGAL provides no unique ID for an event, the OUTPUT_TRANSFORMER creates a new ID for each event, consisting of the gene name, start and end coordinate. The strand and chromosome information are taken from the gtf file.

2.1.4 Whippet -w

Whippet [15] is a tool in the julia programming language to model and quantify alternative splicing events. Whippet generates one output *.psi* file, the columns are explained in 2.4. For each node in the splice graph of a gene, Whippet creates one line in its output file; each node then gets an event type (or NA).The OUTPUT_TRANSFORMER first scans all lines until a new gene appears and then handles all events it found in the previous lines as follows:

IR

Values of Coords column can be used without recalculation.

ES

If the Exc_Paths column has no "NA" value and the Type is "CE", one ES event is created. Additionally the Edges column is checked: If it contains an edge, that covers more than 1 ES event, this line is instead counted as a MES event (see figure 2.3).

column-name	exlpanation
Gene	gene name
Node	Node in splice-graph of gene
Coord	Chromsome and coordinates of this node
Strand	+ or - for strandness of node
Type	Whippet event type abbreviations
[...]	columns for PSI and coverage values
Inc_Paths	Pattern of paths entering this node
Exc_Paths	Pattern of paths exiting this node
Edges	Names of edges passing through this node (edge is represented as node to node)

Table 2.4: Columns of the Whippet output file; Whippet uses a different naming convention than the described "standard" events in 1 (see supplementary table 5.4 for a translation; Whippet has some more detailed types, but they did not occur at all during testing Whippet)

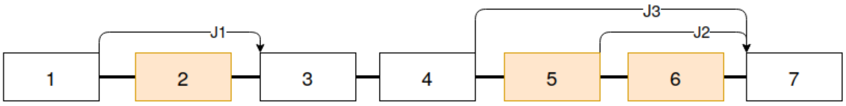

Figure 2.3: node representation in Whippet: each box represents on node in the splice graph. The horizontal bold lines are edges of type n to n+1. Out of J1, J2 and J3 only J3 is considered as a "long_junction". In this case OUTPUT_TRANSFORMER will generate one ES event with the coordinates of box 2 as skipped exon, and one MES event with the coordinates of box 5 and 6 as skipped exons.

A3+ and A5-

If the `Type` column has a value corresponding to A3 or A5, a new event with this type is created with the `Coords` column. In this case, the end coordinate will be increased by 1.

A3- and A5+

If the `Type` column has a value corresponding to A3 or A5, a new event with this type is created with the `Coords` column. In this case, the start coordinate will be decreased by 1.

2.2 Alternative approach to handle skipped exons

As the last chapter showed, it was not possible to get all seven event types out of each tool. Especially the MES and MEE events proposed a challenge and the different outputs made it sometimes more or less hard to get the correct coordinates. Still these coordinates might not be very precise in every case, which is why the OUTPUT_TRANSFORMER has the option to combine all MES, MEE and ES events into a single event type: ES.

The approach is quite trivial, for each skipped exon of the MES and MEE events (n and 2 respectively) a single new ES event will be created. So for example one MES event with 5 skipped exons will result in 5 separate ES events (see example with 2 separate ES in figure 2.4). The *count* column will be used to keep track of how many new events were created from one MES/MEE event (for MEE events there are always only two new events). So again for the previous example, the five ES events would get the following entries in the *count* column: 1, 2, 3, 4, 5.

This feature can be turned off and on easily with a flag in the command line (see section 2.4 for all possible flags).

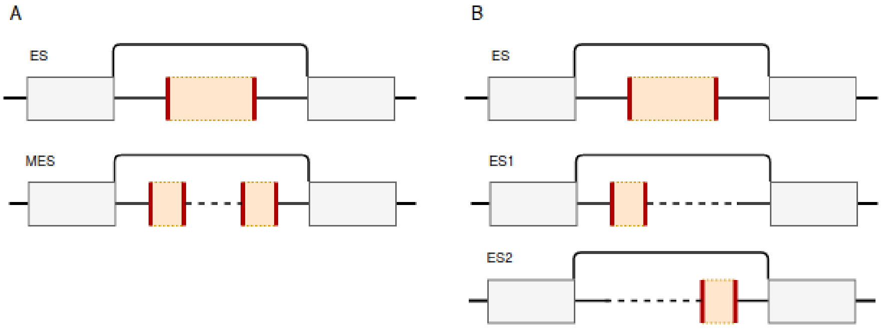

Figure 2.4: **A**: ES and MES are reported as shown in figure 1.2; **B**: the MES event is split up into two separate ES events, ES1 and ES2.

2.3 Comparison with simulated data

Since the overall goal of this project was to allow for easier comparison between tools, a second run mode was implemented, which allows the user to compare the new unified tool output to an event annotation file, where all known events are stored in. For each of the seven event types, this mode will count how many events are correct, by

comparing an event of the tool with every event on the same gene with the same type in the annotation. If the `-strict` flag is used, for this event the start and end coordinate have to be exactly equal; with this flag, only one of them has to be identical. This can be useful when a tool creates many "nan" values for example.
The program also has the option - by using the `-threshold` flag - to set a threshold value for the minimum allowed distance between two events ($distance_{e1,e2} = |e1.start - e2.start| + |e1.end - e2.end|$). Per default this is set to 0, but can be increased to potentially label more events as correct.
Two evaluation scores are calculated: precision and recall. Precision is described as the fraction of correct events divided by the overall number of found events by the tool (equation 2.1) and recall as the fraction of correct events divided by the number of all events in the annotation (equation 2.2). Of course these values are calculated for each event type separately. The raw values as well as the scores can then either be saved in a file or just printed to std-out.

$$precision = \frac{\#correct\ events}{\#found\ events\ by\ tool} \tag{2.1}$$

$$recall = \frac{\#correct\ events}{\#total\ events\ in\ annotation} \tag{2.2}$$

2.4 Documentation and Availability

The OUTPUT_TRANSFORMER has two separate run modes, one to create the unified output file for one (or more tools) and another one to compare this output to an annotation file.

Mode one starts with:

```
output_transformer.py create [-h] [-m MAJIQ_DIR] -s SPLADDER_DIR][-w WHIPPET_FILE]
[-a ASGAL_FILE] -out OUTDIR -gtf GTF [-comb COMBINE_ME]:
```

optional arguments:

-h	-help	show help message and exit
-m	-majiq_dir	directory with 2 majiq output-files: 1) *.psi.tsv (from psi folder) 2) output-file of voila tsv run (named: *voila.tsv)
-s	-spladder_dir	directory with SplAdder output: only *.confirmed.txt files
-w	-whippet_file	whippet-out.psi file
-a	-asgal_file	ASGAL.csv out file
-out	-outdir	output directory
-gtf	-gtf	reference file in gtf format
-comb	-combine_me	Set this to true if you want MES and MEE to be counted as ES events (each skipped exon is one separate ES event)

Table 2.5: possible command line flags for the create runmode

Mode two starts with:

```
output_transformer.py compare [-h] -a EVENT_ANNOTATION -c COMPARE_FILE -gtf
GTF [-stats STATS_OUTFILE] [-comb COMBINE_ME] [-s STRICT] [-t THRESHOLD]:

  optional arguments:

 -h     -help              show help message and exit
 -a     -event_annotation  Event annotation file for ground truth of events
 -c     -compare_file      unified output of AS tool that will be checked
 -comb  -combine_me        Set this to true if you want MES and MEE events to be counted
                           as ES events (each skipped exon is one separate ES event);
                           should also been used when creating the compare file!
 -s     -strict            Use this flag if you want strict comparison between
                           the output and event annotation.  Strict means that both
                           start and end coordinate have to be equal so that an
                           event is counted as correct.
 -t     -threshold         set threshold to allow for events with minimum
                           distance < threshold to still be counted as correct;
                           default is 0
```

Table 2.6: possible command line flags for the `compare` runmode; specification for the event annotation file can be found in table 5.3

The code of this tools and a plotting notebook is currently available as a part of a bigger alternative splicing evaluation project on gitlab: `https://gitlab.lrz.de/ge46ban/dockers`. Also a Snakemake script is placed there to reproduce the results with a given dataset.

3 Results

To test each tool, the R-package ASimulator [17] was used to create three simulated RNA-seq datasets with 50, 100 and 200 million short reads, where each transcript has a combination of two standard event types (for example ES and A3), which are allowed to overlap. Also a simulated sequencing error rate of 0.1 was used. For ASGAL only the 50M read dataset could be calculated due to extreme runtime increases with more sequencing depth (several days).

3.1 Proposing a standard file-format for Alternative Splicing events

Having a standardized file format for a specific field of study is always helpful, but can pose some challenges. Since different tools have different approaches on detecting alternative splicing events (LSVs of MAJIQ vs the "standard" event types in SplAdder and ASGAL), it can get confusing for users to know which tools perform best on their data. The proposed data format (table 3.1) tries to combine these tools into an easy accessible way.

For the four earlier mentioned tools, the process of unifying their output into the new format only takes a few seconds (for all four tools together about 8.3 seconds on the 50M reads dataset and 9.0 seconds on the 200M reads dataset). This means it can easily be applied in a pipeline directly after a tool has finished.

The format also enables research on the coordinate accuracy of each tool, which might not be possible in an easy way with the regular output, due to different ways of annotating events (exon based vs intron based).

3.2 Performance of selected Alternative Splicing tools

In order to evaluate the performance of each tool, precision and recall were calculated using the `output-transformer compare` mode on the unified results of the different datasets.

column	input
chr	symbol of chromosome for this event
gene	gene name for this event
id	unique identifier for this event
strand	+ or −
event_type	one of the following types: ES, IR, A3, A5, ALE, AFE, MEE, MES
count	default=1; can be used for tools like MAJIQ, which report multiple events for one ID to keep track of the number of events; *count* in combination with *id* has to be unique
start_coordinates	one or more start coordinates
end_coordinates	one or more end coordinates

Table 3.1: Allowed inputs for each column in the unified output file

3.2.1 Comparing the combined and separate run mode

These metric allow for a first comparison of the different mode of handling skipped exon events (figure 3.1). For this comparison the 50M reads dataset was used, as well as a distance cutoff of 0 and no strict event comparison.For MAJIQ and Whippet the combined approach has a positive impact regarding the precision values of ES events, with an increase from 0.168 to 0.730 and from 0.265 to 0.904 respectively (marked in red in figure 3.1). Since there are no MEE events calculated for these tools, this increase is only due to the addition of MES events. But while for MAJIQ the increase in precision is connected to a decrease in recall (0.357 to 0.172), Whippet keeps its difference in recall only to about 0.02. This indicates a much stronger detection algorithm.

To summarize, there is a large tradeoff between both modes for MAJIQ tools, but only a positive effect for Whippet. For ASGAL and SlpAdder, this tradeoff is not that big, with only a slight drop in recall, while the precision stays on a very good value - ASGAL only annotates 5 wrong ES events out of its 12890 found events for the combined mode on only misses a single one for the separate mode (1843 out of 1844). SplAdder also has a precision of 0.998 for the combined mode with 7053 correct events out of 7065 found ones.

Overall a big difference in the performance of each tool on each event type can be found; IR events for example are extremely well annotated by Whippet but MAJIQ and SplAdder have their worst overall performance on this type. Also A3 and A5 events are annotated best by Whippet; it is apparent, that each tool has very similar performance values on these two events in itself. The good performance of Whippet can be traced back to its different approach on detecting events, by only relying on the genome annotation and not using an augmented annotation like the three other tools.

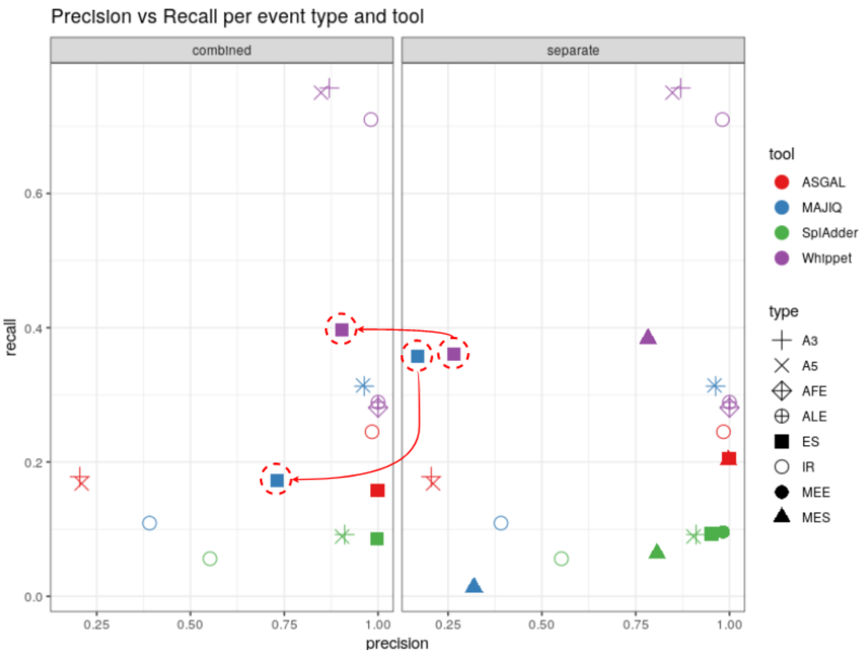

Figure 3.1: Comparing precision and recall for each event type between the different tools. Left shows the "combined" approach, where the event types MES, MEE and ES are all combined into the single ES type. The right plot shows the "separate" mode, were each event type will be counted separately. For better comparison, these three event types are displayed as filled shapes.

3.2.2 Effect of sequencing depth

As a next performance evaluation the three simulated sequencing depths were compared, also taking a look at the two approaches discussed above. Here the "loose" comparison was used (only one coordinate has to be equal) and a distance cutoff of 0 was applied. No tool has outstanding increases in performance, but they generally increase recall for all tools and event types. As mentioned before, ASGAL only has the single datapoint for 50M reads. The precision values stay mostly the same for all tools, no big increases and decreases can be detected. This shows, that generally each tool benefits by increasing the sequencing depth (if they can handle it) and they find more correct events.

The number of found events can be seen in figure 3.3. Is is interesting to look at the difference between MAJIQ and Whippet on the ES events here: when combining events,

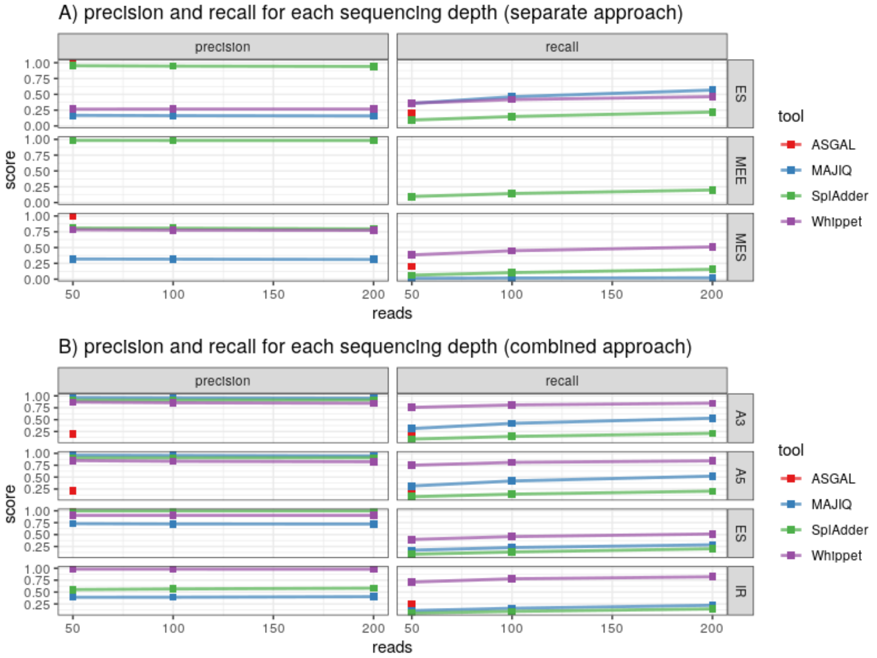

Figure 3.2: Comparing precision and recall for three different sequencing depths. A) shows this for the separate approach for the three affected event types. B) shows the combined approach for the 4 most common event types. ALE and AFE were excluded from this plot, since they are only detected by Whippet; so are MEE events: only SplAdder finds them. For a full view on all types see the supplementary figure 5.3.

Whippet finds about 15000 more events, but when keeping ES separate from MES, Whippet has about 7000 more ES events. This difference can be explained with the MES events. Whippet reports about 6000 of those for the 200M reads dataset, MAJIQ only about 2/3 of this number. So the MES events of Whippet contain much more ES events than the ones of MAJIQ. This can also explain the poor precision and recall values for the MES events of MAJIQ in figure 3.1.

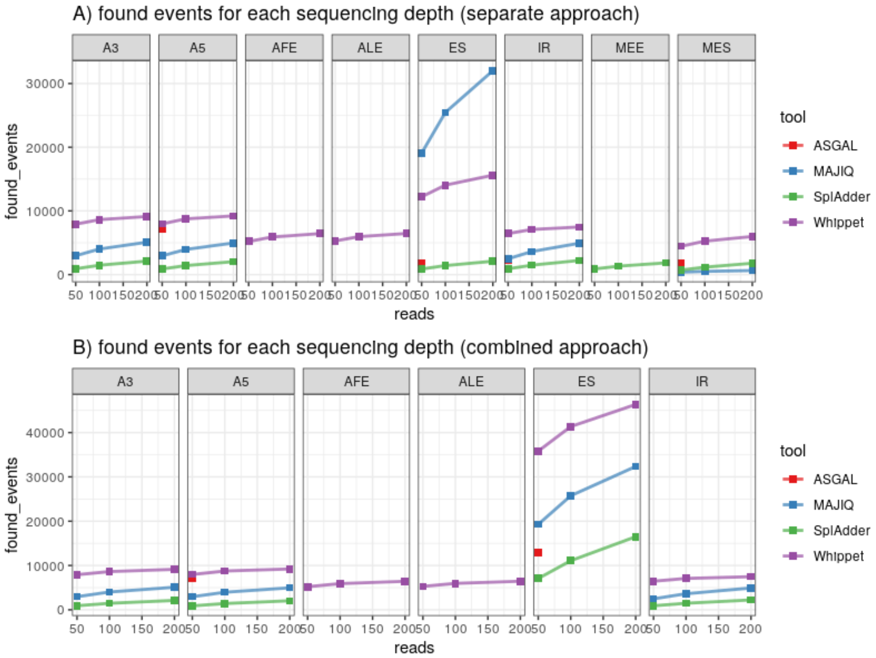

Figure 3.3: Number of found events per sequencing depth

3.2.3 Accuracy of output transformation

In order to confirm that the output was indeed transformed correctly into the unified format, the two parameters `-threshold` and `-strict` were used. For the distance threshold four different values were chosen: 0,1,5,20. Here only the four most frequent event types were compared: ES, A3, A5 and IR. The loose and strict values can have the most impact on performance evaluation, as seen in supplementary figure 5.4. For ASGAL some loss of correct A3 and A5 events can be explained by a small +/-1 issue: in some cases where the first ASGAL junction coordinate is used as start or the second ASGAL junction is used as stop coordinate, this value is off by 2 or 1 positions. This is already accounted for by subtracting an offset of 1 from the first and adding 1 to the second value in these cases, but is cannot catch those events, which are off by a value of 2. Changing that offset to 2 would in return loose all those events which are off by only 1 and are far greater in number (as seen in the different bar heights of A3/A5 with strict comparison and a cutoff of 0 or 1). If one does not care about exact matching of events when comparing, it makes sense to increase the cutoff value for ASGAL.

4 Discussion

The biggest challenges of this project were located in finding the correct transformation of each edge case of the tool outputs. For example the "nan" values of MAJIQ in combination with their way of reporting multiple events in one LSV was not easy to transform. But still this format can be quite useful for easy comparison of tools, as this report tried to show.

4.1 Possible extensions and applications

For now this unified format transformation is only applied to four tools. For other tools it might be faster to implement though, since some already do output their detected events in quite a similar format. This would mean that this format can then be used to calculate large scale benchmarking tests, to compare performance and accuracy of many alternative splicing event detection tools.

4.2 Potential downfalls of the file format

Since each tool has its own way of reporting events, this transformation approach might loose some biological context, that especially MAJIQ and Whippet, with their way of reporting connected events together, try to account for. The unified format still has the *count* column, which can be used to annotate those connections in a way. But since it currently also used for example for the combined approach, to keep track of how many ES events were created by one MES event, these two applications of the column might interfere at some point. Of course one could simply add another column for that case, but then with each additional column, the file format gets more complex and harder to compare again. So its a tradeoff between transportation of as much information as possible, while keeping the file format easily readable and comparable.

5 Supplementary

event-type	strand	start-coordinates	end-coordinates
ES	+/-	start of skipped exon	end of skipped exon
IR	+/-	start of retained intron (previous exon-end +1)	end of retained intron (next exon -1)
A5	+ -	alternative exon-end regular exon-start	regular exon-end alternative exon-start
A3	+ -	regular exon start alternative exon end	alternative exon start regular exon end
AFE	+/-	start of alternative first exon	end of alternative first exon
ALE	+/-	start of alternative last exon	end of alternative last exon
MEE	+/-	start of exclusive exon 1, start of exon exclusive exon 2	end of exclusive exon 1 end of exclusive exon2
MFE	+/-	start of skipped exon 1, start of exon skipped exon 2, ..., start of exon skipped exon n	end of skipped exon1 end of skipped exon 2, ..., end of skipped exon n

Table 5.1: genome coordinates for each event type

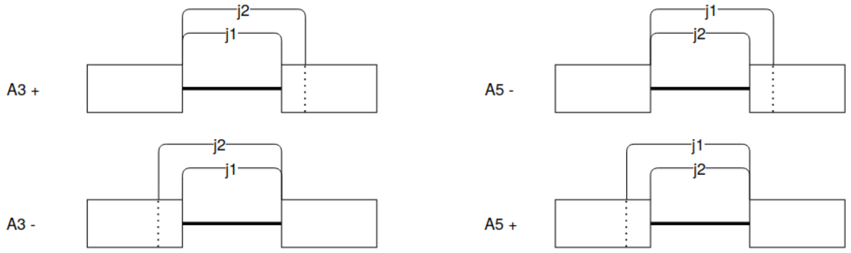

Figure 5.1: schematic view of A3 and A5 events for both strands; + means the strand from 5′ to 3′, - means from 3′ to 5′. Boxes represent exons, bold horizontal lines the intron in between. The dashed line represents the alternative splice junction, j1 and j2 represent the 2 possible splice junctions.

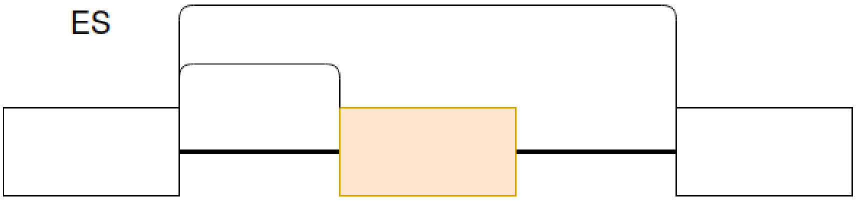

Figure 5.2: visualization of a MAJIQ LSV with a single skipped exon; the two lines show the junction coordinates given by the MAJIQ output. This shows that an exact annotation of the skipped exon (orange box) is not possible with the junction coordinates, at least one coordinate will be missing.

column-name	explanation
`*.psi.tsv file`	
Gene ID	gene name
LSV ID	unique ID for LSV
LSV Type	MAJIQs intern LSV type representation
[multiple PSI-value columns]	[...]
A5SS	`TRUE or FALSE`: if LSV contains one or more A5 events
A3SS	`TRUE or FALSE`: if LSV contains one or more A3 events
ES	`TRUE or FALSE`: if LSV contains one or more ES events
Num. Junctions	Number of junctions in LSV
Num. Exons	Number of exons in LSV
Junctions coords	list of junction coordinates; each list element is a junctions with start and stop coordinate
IR coords	start and stop coordinates of IR event(s)
`voila.tsv file`	
gene_id	gene name
lsv_id	unique ID for LSV
[multiple PSI-value columns]	[...]
lsv_type	MAJIQs intern LSV type representation
num_junctions	Number of junctions in LSV
num_exons	Number of exons in LSV
de_novo_junctions	number of junctions found de novo
seqid	label of chromosome
strand	+ or - for strandness of LSV
junctions_coords	list of junction coordinates; each list element is a junctions with start and stop coordinate
exons_coords	list of exon coordinates;
ir_coords	start and stop coordinates of IR event(s)

Table 5.2: column descriptions for MAJIQ *.psi.tsv file and VOILA voila.tsv file

column-name	explanation
event_annotation	event type
variant	gene name and event types of this variant
template	gene name and event types of this variant
genomic_start	start coordinate of event
genomic_end	end coordinate of event
transcriptomic_start	start coordinate in transcript of event
transcriptomic_end	end coordinate in transcript of event

Table 5.3: Column names and explanation for event annotation file

Whippet type	standard type
CE	ES
AA	A3
AD	A5
RI	IR
TS	-
TE	-
AF	AFE
AL	ALE
BS	-

Table 5.4: Whippet event type to standard event type translation

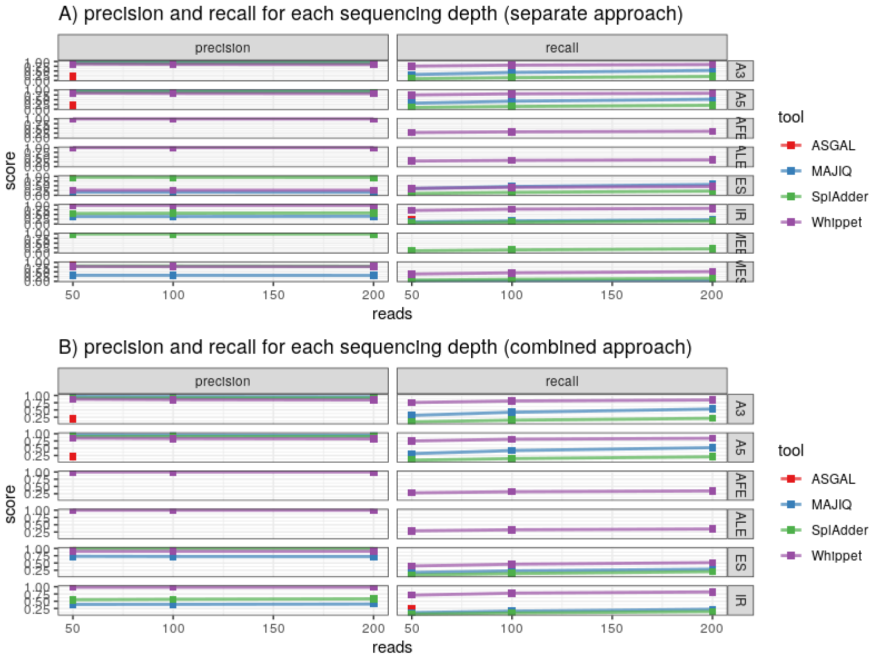

Figure 5.3: Supplementary figure to figure 3.2 with missing event types

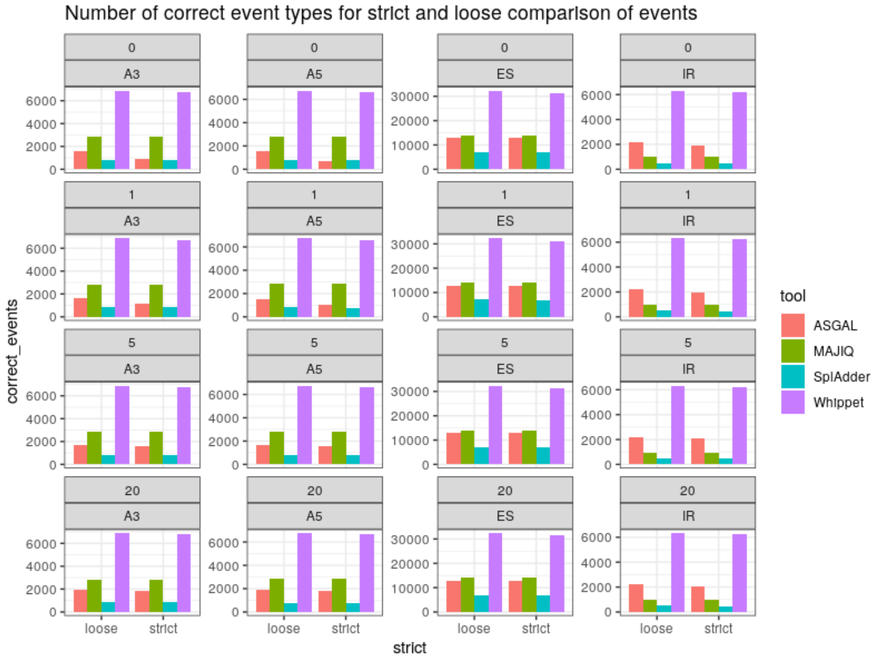

Figure 5.4: Number of correct events for different parameter combinations. Note the different scale in the y axis for ES events, since they appear much more often

List of Figures

List of Tables

Bibliography

[1] T. W. Nilsen and B. R. Graveley. "Expansion of the eukaryotic proteome by alternative splicing." In: *Nature* 463.7280 (Jan. 2010), pp. 457–463. ISSN: 1476-4687. DOI: 10.1038/nature08909. URL: https://doi.org/10.1038/nature08909.

[2] R. K. Singh and T. A. Cooper. "Pre-mRNA splicing in disease and therapeutics." eng. In: *Trends in molecular medicine* 18.8 (Aug. 2012). S1471-4914(12)00101-3[PII], pp. 472–482. ISSN: 1471-499X. DOI: 10.1016/j.molmed.2012.06.006. URL: https://doi.org/10.1016/j.molmed.2012.06.006.

[3] Y. Wang, J. Liu, B. O. Huang, Y.-M. Xu, J. Li, L.-F. Huang, J. Lin, J. Zhang, Q.-H. Min, W.-M. Yang, and X.-Z. Wang. "Mechanism of alternative splicing and its regulation." eng. In: *Biomedical reports* 3.2 (Mar. 2015). br-03-02-0152[PII], pp. 152–158. ISSN: 2049-9434. DOI: 10.3892/br.2014.407. URL: https://doi.org/10.3892/br.2014.407.

[4] Wikipedia. *Alternative splicing — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Alternative%20splicing&oldid=962571544. [Online; accessed 30-August-2020]. 2020.

[5] N. A. Faustino. "Pre-mRNA splicing and human disease." In: *Genes & Development* 17.4 (Feb. 2003), pp. 419–437. DOI: 10.1101/gad.1048803. URL: https://doi.org/10.1101/gad.1048803.

[6] A. Kahles, C. S. Ong, Y. Zhong, and G. Rätsch. "SplAdder: identification, quantification and testing of alternative splicing events from RNA-Seq data." In: *Bioinformatics* 32.12 (Feb. 2016), pp. 1840–1847. DOI: 10.1093/bioinformatics/btw076. URL: https://doi.org/10.1093/bioinformatics/btw076.

[7] L. Denti, R. Rizzi, S. Beretta, G. D. Vedova, M. Previtali, and P. Bonizzoni. "ASGAL: aligning RNA-Seq data to a splicing graph to detect novel alternative splicing events." In: *BMC Bioinformatics* 19.1 (Nov. 2018). DOI: 10.1186/s12859-018-2436-3. URL: https://doi.org/10.1186/s12859-018-2436-3.

[8] M. Estefania, R. Andres, I. Javier, Y. Marcelo, and l. C. Arie. "ASpli: An integrative R package for analysing alternative splicing using RNA-Seq." In: (2020).

[9] W. Wu, J. Zong, N. Wei, J. Cheng, X. Zhou, Y. Cheng, D. Chen, Q. Guo, B. Zhang, and Y. Feng. “CASH: a constructing comprehensive splice site method for detecting alternative splicing events.” In: *Briefings in Bioinformatics* 19.5 (Apr. 2017), pp. 905–917. ISSN: 1477-4054. DOI: 10.1093/bib/bbx034. eprint: https://academic.oup.com/bib/article-pdf/19/5/905/25861142/bbx034.pdf. URL: https://doi.org/10.1093/bib/bbx034.

[10] J. P. Romero, A. Muniategui, F. J. De Miguel, A. Aramburu, L. Montuenga, R. Pio, and A. Rubio. “EventPointer: an effective identification of alternative splicing events using junction arrays.” In: *BMC Genomics* 17.1 (June 2016), p. 467. ISSN: 1471-2164. DOI: 10.1186/s12864-016-2816-x. URL: https://doi.org/10.1186/s12864-016-2816-x.

[11] R. Middleton, D. Gao, A. Thomas, B. Singh, A. Au, J. J.-L. Wong, A. Bomane, B. Cosson, E. Eyras, J. E. J. Rasko, and W. Ritchie. “IRFinder: assessing the impact of intron retention on mammalian gene expression.” In: *Genome Biology* 18.1 (Mar. 2017), p. 51. ISSN: 1474-760X. DOI: 10.1186/s13059-017-1184-4. URL: https://doi.org/10.1186/s13059-017-1184-4.

[12] G. A. Sacomoto, J. Kielbassa, R. Chikhi, R. Uricaru, P. Antoniou, M.-F. Sagot, P. Peterlongo, and V. Lacroix. “KIS SPLICE: de-novo calling alternative splicing events from RNA-seq data.” In: *BMC Bioinformatics* 13.6 (Apr. 2012), S5. ISSN: 1471-2105. DOI: 10.1186/1471-2105-13-S6-S5. URL: https://doi.org/10.1186/1471-2105-13-S6-S5.

[13] S. S. Norton, J. Vaquero-Garcia, N. F. Lahens, G. R. Grant, and Y. Barash. “Outlier detection for improved differential splicing quantification from RNA-Seq experiments with replicates.” In: *Bioinformatics* 34.9 (Dec. 2017). Ed. by B. Berger, pp. 1488–1497. DOI: 10.1093/bioinformatics/btx790. URL: https://doi.org/10.1093/bioinformatics/btx790.

[14] L. D. Goldstein, Y. Cao, G. Pau, M. Lawrence, T. D. Wu, S. Seshagiri, and R. Gentleman. “Prediction and Quantification of Splice Events from RNA-Seq Data.” In: *PLOS ONE* 11.5 (May 2016), pp. 1–18. DOI: 10.1371/journal.pone.0156132. URL: https://doi.org/10.1371/journal.pone.0156132.

[15] T. Sterne-Weiler, R. J. Weatheritt, A. J. Best, K. C. Ha, and B. J. Blencowe. “Efficient and Accurate Quantitative Profiling of Alternative Splicing Patterns of Any Complexity on a Laptop.” In: *Molecular Cell* 72.1 (Oct. 2018), 187–200.e6. DOI: 10.1016/j.molcel.2018.08.018. URL: https://doi.org/10.1016/j.molcel.2018.08.018.

[16] J. Vaquero-Garcia, A. Barrera, M. R. Gazzara, J. González-Vallinas, N. F. Lahens, J. B. Hogenesch, K. W. Lynch, and Y. Barash. "A new view of transcriptome complexity and regulation through the lens of local splicing variations." In: *eLife* 5 (Feb. 2016). Ed. by J. Valcárcel, e11752. ISSN: 2050-084X. DOI: `10.7554/eLife.11752`. URL: `https://doi.org/10.7554/eLife.11752`.

[17] Q. Manz. *ASimulator*. `https://github.com/biomedbigdata/ASimulatoR`. 2020.