AS-Quant User Manual

Naima Ahmed Fahmi, Heba Nassereddeen, Jae-Woong Chang, Meeyeon Park, Hsin-Sung Yeh, Jiao Sun, Deliang Fan, Dylan Vidal, Jeongsik Yong and Wei Zhang

1. About

AS-Quant is a computational tool used to detect alternative splicing (AS) events of two biological conditions- specifically, two groups of samples- from RNA-seq data. It can categorize five major types of AS in a comparative and comprehensive manner. AS-Quant also includes a visualization tool which generates plots for both the AS events and the annotation of the whole gene.

2. Download

AS-Quant tool can be downloaded directly from Github . Users need to have Python installed on their machine. It can work on both Windows and Linux platforms.

3. Required tools

a. Python (version 3.0 or higher)
b. Samtools 0.1.8* [This specific version, provided with the code]

Required python packages

- matplotlib
  [$sudo apt-get install python3-matplotlib **or** $pip install matplotlib]
- scipy
  [$sudo apt-get install python3-scipy **or** $pip install scipy]
- pandas
  [$sudo apt-get install python3-pandas **or** $pip install pandas]

4. Run AS-Quant

AS-Quant is designed to work on both human (hg19/hg38) and mouse (mm10) alternative splicing events. The supplementary data (the five types of alternative splicing exons' dataset and the reformatted annotation) is provided in the project directory in Github.

Users have to run the following two Python files in order to run AS-Quant:

a. **as_quant.py**: the main function which the user needs to run.
b. **make_plots.py**: generates figures for visual representation of data.

4. 1. Run as_quant.py

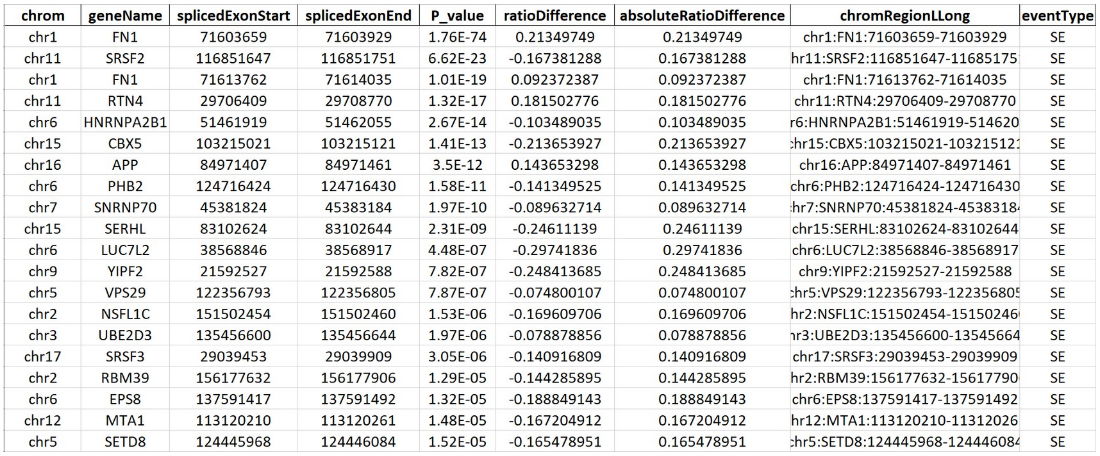
Command: **$ python3 as_quant.py [options]**
AS-Quant supports the analysis of multiple samples or replicates (in .bam format) in each group/condition. For eliminating the false positives and ignoring the noises of the samples, **it is recommended to provide input data with at least 50M reads on each sample.**

Users can also select the underlying method [-method option] to determine the significance of each alternative splicing event. Available methods are: chi-squared test (chisquare) and Wilcoxon rank-sum test (ranksum). If there is more than one sample or replicate in each group, Wilcoxon rank-sum test is an additional option to determine the significance of the events.

List of available options
(** refers to a mandatory field)*

-i*	Input directories. The two directories for the two groups of samples in consecutive order.
-s*	Species name: hg38 or hg19 for human, mm10 for mouse
-c	Number of cores to utilize. Default is to run sequentially on one core. Users can also use the MAX flag to utilize all available cores.
-o	Output directory
-novel	If the user wants to detect both novel and annotated alternative splicing events.Default is annotated only.
-method	Statistical method to determine the significance of the potential alternative splicing events: chisquare/ranksum. Default is chisquare.

as-quant.py will generate several intermediary files in the output directory. After computing the significance of the association between the two conditions, the final results will be written in the spreadsheet named **asquant_group1_vs_group2.xlsx**, with five separate sheets for five different splicing types. The following image shows some of the generated fields in **asquant_group1_vs_group2.xlsx** for event type 'SE':



chrom	geneName	splicedExonStart	splicedExonEnd	P_value	ratioDifference	absoluteRatioDifference	chromRegionLLong	eventType
chr1	FN1	71603659	71603929	1.76E-74	0.21349749	0.21349749	chr1:FN1:71603659-71603929	SE
chr11	SRSF2	116851647	116851751	6.62E-23	-0.167381288	0.167381288	hr11:SRSF2:116851647-11685175	SE
chr1	FN1	71613762	71614035	1.01E-19	0.092372387	0.092372387	chr1:FN1:71613762-71614035	SE
chr11	RTN4	29706409	29708770	1.32E-17	0.181502776	0.181502776	chr11:RTN4:29706409-29708770	SE
chr6	HNRNPA2B1	51461919	51462055	2.67E-14	-0.103489035	0.103489035	r6:HNRNPA2B1:51461919-514620	SE
chr15	CBX5	103215021	103215121	1.41E-13	-0.213653927	0.213653927	hr15:CBX5:103215021-10321512	SE
chr16	APP	84971407	84971461	3.5E-12	0.143653298	0.143653298	chr16:APP:84971407-84971461	SE
chr6	PHB2	124716424	124716430	1.58E-11	-0.141349525	0.141349525	chr6:PHB2:124716424-124716430	SE
chr7	SNRNP70	45381824	45383184	1.97E-10	-0.089632714	0.089632714	hr7:SNRNP70:45381824-4538318	SE
chr15	SERHL	83102624	83102644	2.31E-09	-0.24611139	0.24611139	chr15:SERHL:83102624-83102644	SE
chr6	LUC7L2	38568846	38568917	4.48E-07	-0.29741836	0.29741836	chr6:LUC7L2:38568846-38568917	SE
chr9	YIPF2	21592527	21592588	7.82E-07	-0.248413685	0.248413685	chr9:YIPF2:21592527-21592588	SE
chr5	VPS29	122356793	122356805	7.87E-07	-0.074800107	0.074800107	hr5:VPS29:122356793-12235680	SE
chr2	NSFL1C	151502454	151502460	1.53E-06	-0.169609706	0.169609706	hr2:NSFL1C:151502454-15150246	SE
chr3	UBE2D3	135456600	135456644	1.97E-06	-0.078878856	0.078878856	r3:UBE2D3:135456600-13545664	SE
chr17	SRSF3	29039453	29039909	3.05E-06	-0.140916809	0.140916809	chr17:SRSF3:29039453-29039909	SE
chr2	RBM39	156177632	156177906	1.29E-05	-0.144285895	0.144285895	hr2:RBM39:156177632-15617790	SE
chr6	EPS8	137591417	137591492	1.32E-05	-0.188849143	0.188849143	chr6:EPS8:137591417-137591492	SE
chr12	MTA1	113120210	113120261	1.48E-05	-0.167204912	0.167204912	hr12:MTA1:113120210-11312026	SE
chr5	SETD8	124445968	124446084	1.52E-05	-0.165478951	0.165478951	hr5:SETD8:124445968-12444608	SE

The column ‘P_value’ defines the significance of the detected events. The results are sorted based on *P*-values. The splicing events with **‘absoluteRatioDifference’>=1** and **P_value < 0.05** are considered to be significant between two conditions. The genes which have the average read coverage < 10 on all the samples, are removed from the analysis. In the ‘Ratio difference’ column, a large positive ratio difference indicates a potential splicing event in group 2 (condition 2), whereas a negative ratio difference with a large absolute value indicates a potential splicing event in group 1 (condition 1).

4.2. Running AS-Quant with provided sample data

We provided sample data ‘sample_input_mouse’ in our GitHub repository to test AS-Quant, where group1 and group2 are two directories containing the input bam files.

Command:

```
$ python3 as_quant.py -s mm10 -i /<dir>/sample_input_mouse/group1 /<dir>/sample_input_mouse/group2 -o sample_output
```

4.3. Run make_plots.py

AS-Quant provides a visualization tool,makeplots.py, which generates plots for both the AS events and the annotation of the whole gene. To run the visualization tool, users need to enter the following command:

```
$ python3 make_plots.py -s species -o outputdirectory -i input1 input2
```

Example:

```
$ python3 make_plots.py -s mm10 -i sample_input_mouse/group1 sample_input_mouse/group2 -o annotation_plot
```

Next, make_plots.py will ask the users to enter the region of interest for which they want to generate the annotation plot. The format is specific: **Chom:GeneName:RegionStart-RegionEnd**

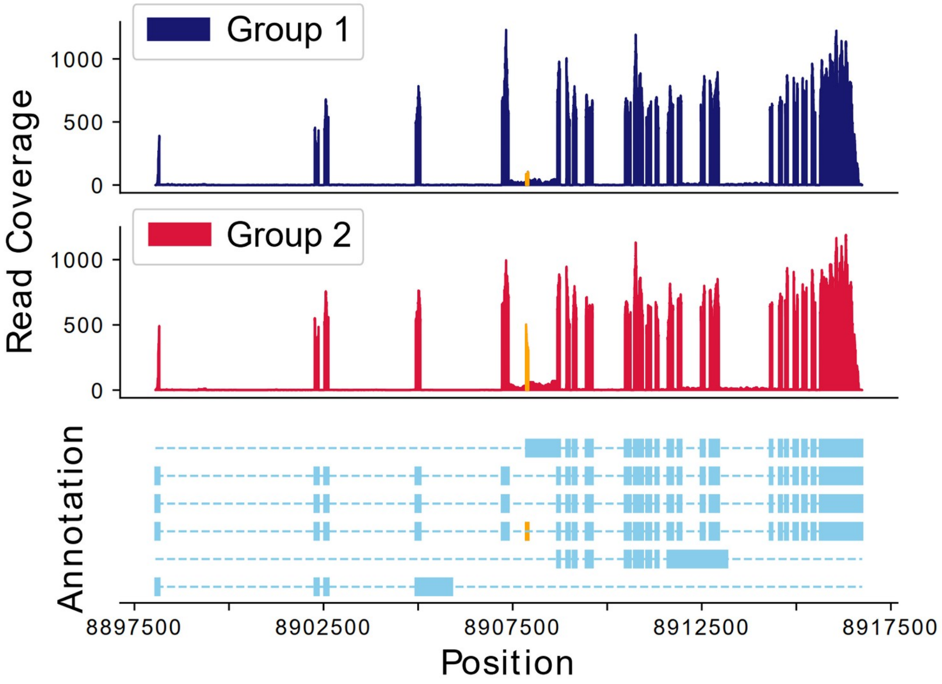
make_plots.py parameter descriptions

Chrom **:**	Name of the chromosome
GeneName **:**	Name of the gene
RegionStart **:**	Starting position of the region
Region End **:**	End position of the region

Example: **chr1:Tceb1:16641724-16643478**

The users can generate multiple plots, one by one, by providing one input region at a time. The code exits when the user puts the text 'Exit' and presses Enter.

Make_plots.py will generate the read coverage plot for the given gene along with the whole annotation plot with all exons' information of that gene. The figure below shows an example of the read coverage plot generated by AS-Quant.



The first two subplots of the figure represent the read coverage of the two biological conditions. The bottom subplot shows the gene annotation and the exon information of that gene.