AS-Quant User Manual

Naima Ahmed Fahmi, Heba Nassereddeen, Jae-Woong Chang, Meeyeon Park, Hsin-Sung Yeh, Jiao Sun, Deliang Fan, 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 the Windows, Linux and Mac platforms.

3. Required tools

a. Python (version 3.0 or higher)
b. Samtools 0.1.8* [This specific version]

Required python packages

- matplotlib
  Example Python command: $sudo apt-get install python3-matplotlib
- scipy
  Example Python command: $ sudo apt-get install python3-scipy
- pandas
  Example Python command: $ sudo apt-get install python3-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 target dataset and the 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]**

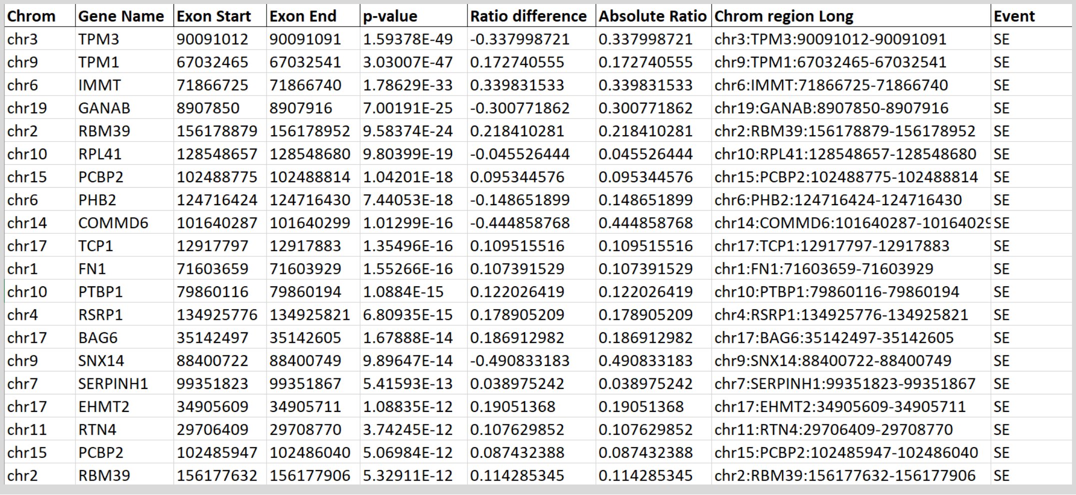
AS-Quant supports the analysis of multiple samples or replicates (in .bam format) in each group. Users can also select the underlying method {[}-method option{]} to determine the significance of each alternative splicing event. Available methods are: chi-squared test (chisquare) and Wilcoxon rank-sum 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
-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 splicingevents: 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 **group1_vs_group2.xlsx**, with five separate sheets for five different splicing types.The following image shows some of the generated fields in **group1_vs_group2.xlsx** for event type ‘SE’:



Chrom	Gene Name	Exon Start	Exon End	p-value	Ratio difference	Absolute Ratio	Chrom region Long	Event
chr3	TPM3	90091012	90091091	1.59378E-49	-0.337998721	0.337998721	chr3:TPM3:90091012-90091091	SE
chr9	TPM1	67032465	67032541	3.03007E-47	0.172740555	0.172740555	chr9:TPM1:67032465-67032541	SE
chr6	IMMT	71866725	71866740	1.78629E-33	0.339831533	0.339831533	chr6:IMMT:71866725-71866740	SE
chr19	GANAB	8907850	8907916	7.00191E-25	-0.300771862	0.300771862	chr19:GANAB:8907850-8907916	SE
chr2	RBM39	156178879	156178952	9.58374E-24	0.218410281	0.218410281	chr2:RBM39:156178879-156178952	SE
chr10	RPL41	128548657	128548680	9.80399E-19	-0.045526444	0.045526444	chr10:RPL41:128548657-128548680	SE
chr15	PCBP2	102488775	102488814	1.04201E-18	0.095344576	0.095344576	chr15:PCBP2:102488775-102488814	SE
chr6	PHB2	124716424	124716430	7.44053E-18	-0.148651899	0.148651899	chr6:PHB2:124716424-124716430	SE
chr14	COMMD6	101640287	101640299	1.01299E-16	-0.444858768	0.444858768	chr14:COMMD6:101640287-1016402	SE
chr17	TCP1	12917797	12917883	1.35496E-16	0.109515516	0.109515516	chr17:TCP1:12917797-12917883	SE
chr1	FN1	71603659	71603929	1.55266E-16	0.107391529	0.107391529	chr1:FN1:71603659-71603929	SE
chr10	PTBP1	79860116	79860194	1.0884E-15	0.122026419	0.122026419	chr10:PTBP1:79860116-79860194	SE
chr4	RSRP1	134925776	134925821	6.80935E-15	0.178905209	0.178905209	chr4:RSRP1:134925776-134925821	SE
chr17	BAG6	35142497	35142605	1.67888E-14	0.186912982	0.186912982	chr17:BAG6:35142497-35142605	SE
chr9	SNX14	88400722	88400749	9.89647E-14	-0.490833183	0.490833183	chr9:SNX14:88400722-88400749	SE
chr7	SERPINH1	99351823	99351867	5.41593E-13	0.038975242	0.038975242	chr7:SERPINH1:99351823-99351867	SE
chr17	EHMT2	34905609	34905711	1.08835E-12	0.19051368	0.19051368	chr17:EHMT2:34905609-34905711	SE
chr11	RTN4	29706409	29708770	3.74245E-12	0.107629852	0.107629852	chr11:RTN4:29706409-29708770	SE
chr15	PCBP2	102485947	102486040	5.06984E-12	0.087432388	0.087432388	chr15:PCBP2:102485947-102486040	SE
chr2	RBM39	156177632	156177906	5.32911E-12	0.114285345	0.114285345	chr2:RBM39:156177632-156177906	SE

4.2. Running AS-Quant with provided sample input

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 asquant.py -s mouse -i sample_input_mouse/group1 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 mouse -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 should be specific:

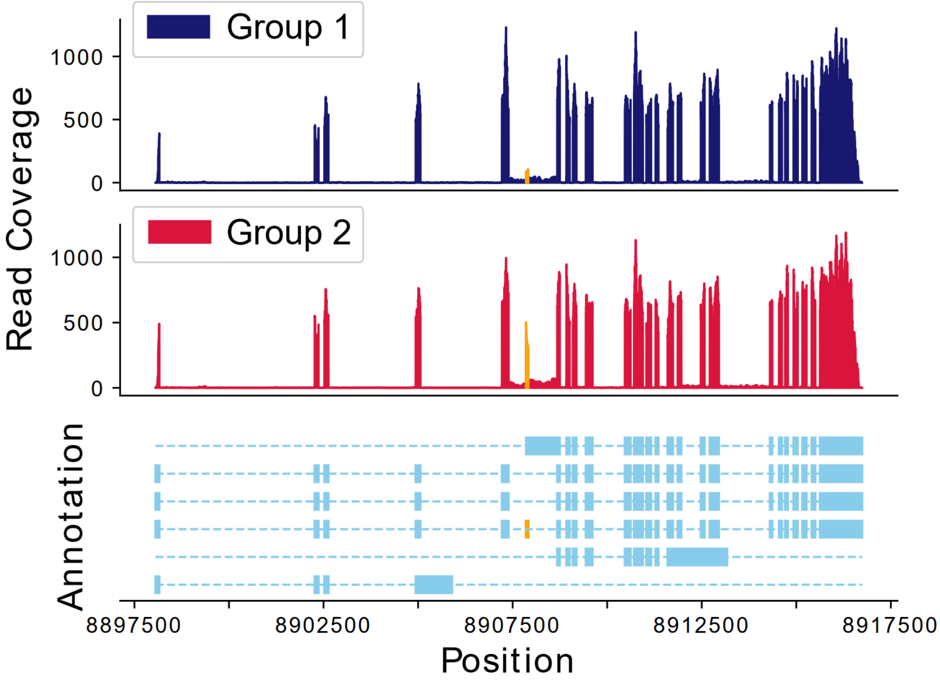
Chom:GeneName:RegionStart-RegionEnd

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**

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