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

4. **Required python packages**
   a. matplotlib
      Example Python command: $sudo apt-get install python3-matplotlib
   b. scipy
      Example Python command: $ sudo apt-get install python3-scipy
   c. pandas
      Example Python command: $ sudo apt-get install python3-pandas

5. **Running AS-Quant**

   AS-Quant is designed for handling both human and mouse alternative splicing events. The supplementary data (the five types of alternative splicing target dataset and the annotation) is provided in the project directory.

   Users have to run the following two Python files in order to run AS-Quant:
   i) as_quant.py: the main function which the user need to run
   ii) make_plots.py: generates figures for visual representation of data

The software comes with a default configuration file which allows the user to specify 1) the directories of the input samples, 2) the species to be analyzed, and 3) the directory of the folder where all output files will be stored. AS-Quant supports the analysis of multiple samples that belong to two different groups- all BAM files inside the input1 directory will be considered as part of the first group, and all BAM files inside the input2 directory will be considered as part of the second group. It is required to have at least one BAM file in each input directory.

Running Parameters in the configuration.ini file: (* refers to a mandatory field)

species*:	Species name (human/mouse)
output_dir:	Output directory. Users can specify the desired output directory for writing the results. [Optional]
input1* **:**	Specifies the directory containing the first group of samples [must be a folder name **without** '/' at the end]
input2* **:**	Specifies the directory containing the second group of samples [must be a folder name **without** '/' at the end.]

Once the running parameters have been specified, the user should save the configuration.ini file, open a terminal and enter the following command to run AS-Quant:

```
$ python3  as_quant.py
```

as_quant.py will generate several intermediary files in the directory named **Output** (if the user does not provide a new output directory). After computing the significance of the association between the two groups of samples, the final results will be written in the file named **Group1_Vs_Group2.csv**. The following image shows some of the generated fields in **Group1_Vs_Group2.csv**:

	A	B	C	D	E	F	G	H
1	Chrom	Gene Name	Exon Start	Exon End	p-value	Ratio difference	Absolute Ratio difference	Chrom region
2	chr1	DPH5	101467022	101467100	0.9121302401	0.1000592552	0.1000592552	chr1:DPH5:101467022-101467100
3	chr1	DPH5	101479265	101479374	0.5834973654	0.1790025326	0.1790025326	chr1:DPH5:101479265-101479374
4	chr1	APITD1-CORT	10494713	10494747	0.3359058847	-0.972067033	0.972067033	chr1:APITD1-CORT:10494713-10494747
5	chr1	PEX14	10596269	10596354	0.9023390528	0.137534426	0.137534426	chr1:PEX14:10596269-10596354
6	chr1	PEX14	10659294	10659423	0.6931817062	0.2498379422	0.2498379422	chr1:PEX14:10659294-10659423
7	chr1	AMPD2	110167924	110168055	0.563988963	0.1287839664	0.1287839664	chr1:AMPD2:110167924-110168055
8	chr1	SLC16A4	110924273	110924417	0.5841909446	-0.8237182045	0.8237182045	chr1:SLC16A4:110924273-110924417
9	chr1	SLC16A4	110925455	110925588	0.6872561265	-0.5067973124	0.5067973124	chr1:SLC16A4:110925455-110925588
10	chr1	ST7L	113098489	113098640	0.9287630199	0.196174765	0.196174765	chr1:ST7L:113098489-113098640
11	chr1	ST7L	113140592	113140708	0.7976173158	0.366711548	0.366711548	chr1:ST7L:113140592-113140708
12	chr1	ST7L	113143415	113143470	0.9416810826	-0.1870431467	0.1870431467	chr1:ST7L:113143415-113143470

Running AS-Quant with provided sample input (Optional)

We have sample data available in our GitHub repository that you can use to test AS-Quant**.** To run AS-Quant with the sample data, download the *mm10* and *Sample_input_mouse*

folders and place them in the same directory containing the .py files and Samtools 0.1.8. Next, make the following changes to the configuration.ini file (assuming that AS-Quant is the root directory):

Parameter:	**Value:**
species	mouse
output_dir:	AS-Quant/sample_output
input1* **:**	AS-Quant/Sample_input_mouse/Group1
input2* **:**	AS-Quant/Sample_input_mouse/Group2

After specifying the running parameters, save the configuration.ini file and enter the following command to run AS-Quant with the sample data:

```
$ python3  as_quant.py
```

The above command will generate the output tables inside of a folder called 'sample_output' in the same directory. After computing the significance of the association between the two groups of samples, the final results will be written in the file named **Group1_Vs_Group2.csv**.

6. **Running make_plots.py**

AS Quant provides a visualization tool, makes_plots.py, which generates plots for both the AS events and the annotation of the whole gene. To run the visualization tool, enter the following command inside a terminal:

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

Example:

```
$ python3  make_plots.py  -s  human  -o  Annotation_plot  inputs/group1  inputs/group2
```

Next, make_plots.py will ask the user to enter the region of interest for which they want to generate the annotation plot. The format should be in a specific format:
Chom:GeneName:RegionStart-RegionEnd

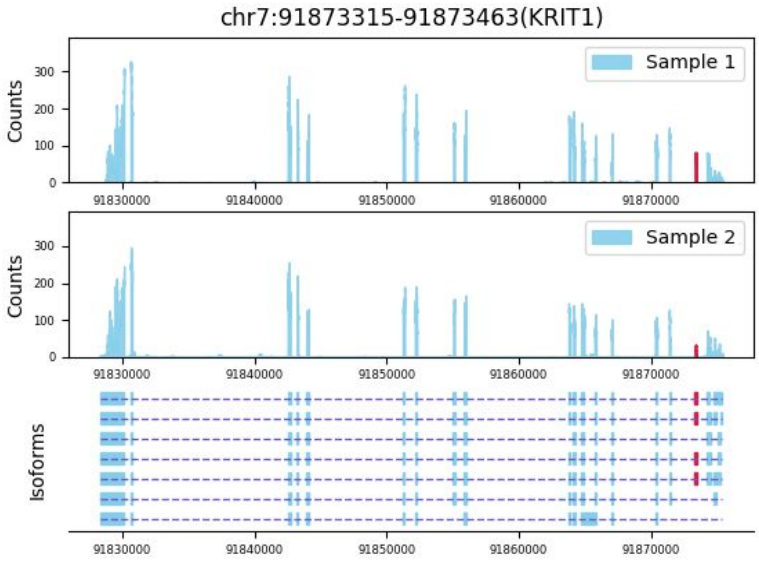
make_plots.py parameter descriptions

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 output will produce a figure like the one below:



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.