

AS-Quant User Manual

Naima Ahmed Fahmi, Jeongsik Yong, Wei Zhang

May, 2019

1. About: AS-Quant is a computational tool used to detect alternative splicing(AS) events from RNA-seq data of two biological conditions (two samples). 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 [AS-Quant](#) . Users need to have python installed on their machine. It can work on the Windows, Linux and Mac platform.

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. Run 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

```
$ python3 as_quant.py -s species -o output_directory input1_dir1
input2_dir2
```

Example:

```
$ python3 as_quant.py -s human -o results home/Naima/input1.bam
home/Naima/input2.bam
```

Options: (* refers to mandatory field)

| | |
|---------------------------|--|
| <code>-s/-S*</code> : | Species name. AS-Quant can handle both human and mouse. |
| <code>-o/-O</code> : | Output directory. User can specify desired output directory for writing the results. [Optional] Output directory must be a folder name/directory without a '/' at the end. |
| <code>input1*</code> : | Specifies directory for the first sample. Input1 is the name of a bam file aligned to reference genome. |
| <code>input2*</code> : | Specifies directory for the second sample. Input2 is the name of a bam file aligned to reference genome. |

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 samples, the final results will be written in the file named **sample1_Vs_sample2.csv**. The following image is showing some of the generated fields in **sample1_Vs_sample2.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 | 0.7976713158 | 0.366711548 | 0.366711548 | chr1:ST7L:113140592-113140708 |
| 12 | chr1 | ST7L | 113143415 | 113143470 | 0.9416810826 | -0.1870431467 | 0.1870431467 | chr1:ST7L:113143415-113143470 |

Run AS-Quant with provided sample input (Optional)

```
$ python3 as_quant.py -s mouse sample_input_mouse/s1/accepted_hits.bam  
sample_input_mouse/s2/accepted_hits.bam
```

It will generate the output tables inside of folder 'Output' in the same directory.
Or you can generate output in your desired directory, such as 'Results':

```
$ python3 as_quant.py -s mouse -o Results  
sample_input_mouse/s1/accepted_hits.bam  
sample_input_mouse/s2/accepted_hits.bam
```

6. Run make_plots.py

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

Example:

```
$ python3 make_plots.py -s human -o Annotation_plot inputs/sample1  
inputs/sample2
```

At that point, 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:

Chrom:GeneName:RegionStart-RegionEnd

Parameter description

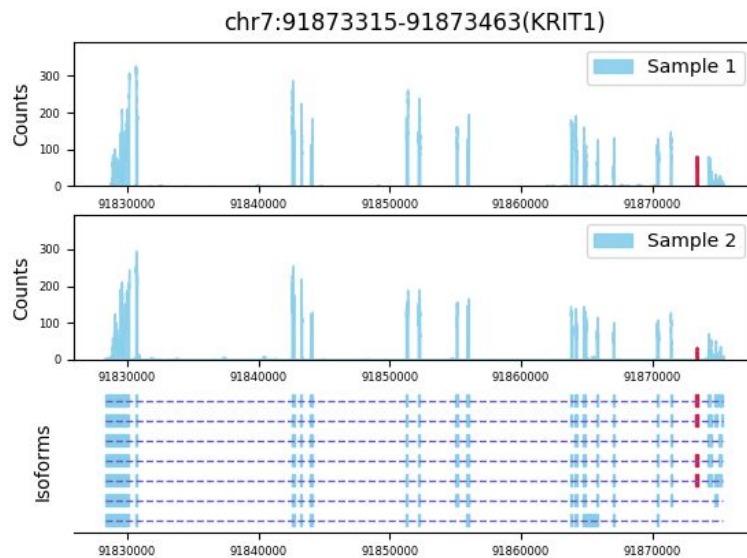
| | |
|---------------|---------------------------------|
| Chrom : | Name of the chromosome |
| GeneName : | Name of the gene. |
| RegionStart : | Starting position of the region |
| Region End : | End position of the region |

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 following:



The first two subplots of the figure represent the read coverage of the two biological conditions. The bottom subplot shows the gene annotation along with all the exons information of that gene.