APA-Scan User Manual

Naima Ahmed Fahmi, Khandakar Tanvir Ahmed, Jae-Woong Chang, Heba Nassereddeen, Deliang Fan, Dylan Vidal, Jeongsik Yong and Wei Zhang

1. About

APA-Scan is a computational tool which can detect and visualize genome-wide 3’-UTR APA events. APA-Scan integrates both 3’-end-seq (an RNA-seq method with a specific enrichment of 3’-ends of mRNA) data and the location information of predicted canonical PASs with RNA-seq data to improve the quantitative definition of genome-wide UTR-APA events. It is also advantageous in producing high quality plots of the user defined events.

2. Download

APA-Scan is downloadable directly from github. Users need to have python (version 3.0 or higher) installed in their machine.

3. Required Softwares

a. Python (v3.0 or higher)
b. Samtools (v 0.1.8)* [This specific version is mandatory]

4. Required python packages (Can install using pip, or other process)

a. Pandas: $ pip install pandas
b. Bio: $ pip install biopython
c. Scipy: $ pip install scipy
d. Numpy: $ pip install numpy
e. Peakutils: pip install PeakUtils

5. Run APA-Scan

APA-Scan can handle both human and mouse data for detecting potential APA truncation sites. The tool is designed to follow the format of Refseq annotation and genome file from UCSC Genome Browser. Users need to have the following two files in the parent directory in order to run APA-Scan:

- Refseq annotation (.txt format)
- Genome fasta file (downloaded from UCSC genome browser)

APA-Scan has two python scripts: APA-Scan.py, Make-Plots.py
And 1 configuration file: configuration.ini

The configuration file 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.

APA-Scan 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.

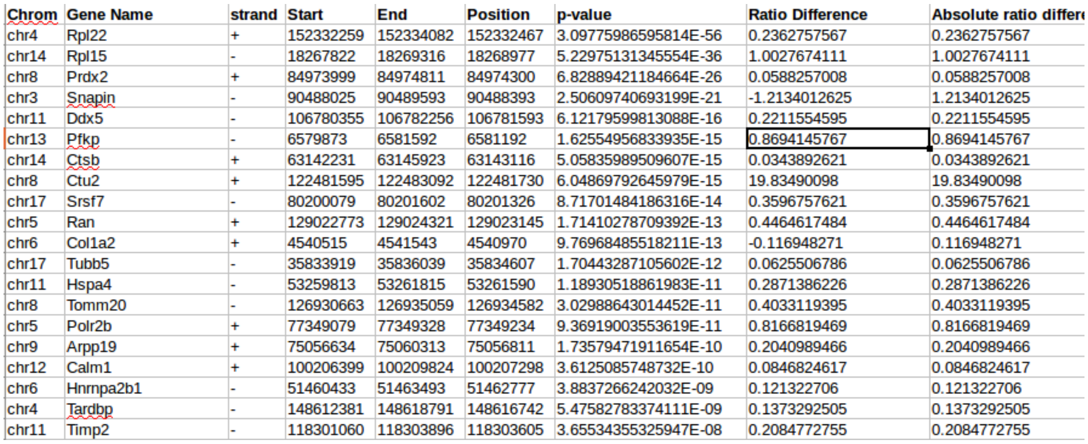
Running Parameters in the configuration.ini file: (* refers to a mandatory field)

species*:	Species name (human/mouse)
input1* **:**	Directory containing the first group of samples with RNA-seq data [must be a folder name **without** '/' at the end]
input2* **:**	Directory containing the second group of samples with RNA-seq data[must be a folder name **without** '/' at the end.]
pas1* **:**	Directory containing the first group of samples with 3'-end-seq data [must be a folder name **without** '/' at the end]. Default is NULL
pas2* **:**	Directory containing the second group of samples with 3'-end-seq data [must be a folder name **without** '/' at the end]. Default is NULL
extended* **:**	APA-Scan will run on `Extended 3UTR' mode and it will search for APA sites upto 10kb downstream of the annotated transcript. Value: yes or no
annotation* **:**	RefSeq annotation file, downloaded from UCSC Genome Browser, in .txt format
genome* **:**	Genome fasta file, in .fa format
cores* **:**	Number of cores to utilize. Input greater than 1 core to run the code in parallel. Default is NULL, which will run the code sequentially. Additionally, the MAX flag uses all available cores.
output_dir **:**	Output directory. Users can specify the desired output directory for writing the results. [Optional]

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 APA-Scan:

```
$ python3   APA-Scan.py
```

APA-Scan.py will generate several intermediary files in the 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:



Chrom	Gene Name	strand	Start	End	Position	p-value	Ratio Difference	Absolute ratio differe
chr4	Rpl22	+	152332259	152334082	152332467	3.09775986595814E-56	0.2362757567	0.2362757567
chr14	Rpl15	-	18267822	18269316	18268977	5.22975131345554E-36	1.0027674111	1.0027674111
chr8	Prdx2	+	84973999	84974811	84974300	6.82889421184664E-26	0.0588257008	0.0588257008
chr3	Snapin	-	90488025	90489593	90488393	2.50609740693199E-21	-1.2134012625	1.2134012625
chr11	Ddx5	-	106780355	106782256	106781593	6.12179599813088E-16	0.2211554595	0.2211554595
chr13	Pfkp	-	6579873	6581592	6581192	1.62554956833935E-15	0.8694145767	0.8694145767
chr14	Ctsb	+	63142231	63145923	63143116	5.05835989509607E-15	0.0343892621	0.0343892621
chr8	Ctu2	+	122481595	122483092	122481730	6.04869792645979E-15	19.83490098	19.83490098
chr17	Srsf7	-	80200079	80201602	80201326	8.71701484186316E-14	0.3596757621	0.3596757621
chr5	Ran	+	129022773	129024321	129023145	1.71410278709392E-13	0.4464617484	0.4464617484
chr6	Col1a2	+	4540515	4541543	4540970	9.76968485518211E-13	-0.116948271	0.116948271
chr17	Tubb5	-	35833919	35836039	35834607	1.70443287105602E-12	0.0625506786	0.0625506786
chr11	Hspa4	-	53259813	53261815	53261590	1.18930518861983E-11	0.2871386226	0.2871386226
chr8	Tomm20	-	126930663	126935059	126934582	3.02988643014452E-11	0.4033119395	0.4033119395
chr5	Polr2b	+	77349079	77349328	77349234	9.36919003553619E-11	0.8166819469	0.8166819469
chr9	Arpp19	+	75056634	75060313	75056811	1.73579471911654E-10	0.2040989466	0.2040989466
chr12	Calm1	+	100206399	100209824	100207298	3.6125085748732E-10	0.0846824617	0.0846824617
chr6	Hnrnpa2b1	-	51460433	51463493	51462777	3.8837266242032E-09	0.121322706	0.121322706
chr4	Tardbp	-	148612381	148618791	148616742	5.47582783374111E-09	0.1373292505	0.1373292505
chr11	Timp2	-	118301060	118303896	118303605	3.65534355325947E-08	0.2084772755	0.2084772755

6. **Run Make-plots.py**
   Make-plots.py also requires the same configuration file to run. It will use the input and output directories listed in the configuration file and prepare a read coverage plot along with the 3’-UTR annotation based on user defined region.

```
python3  Make-plots.py
```

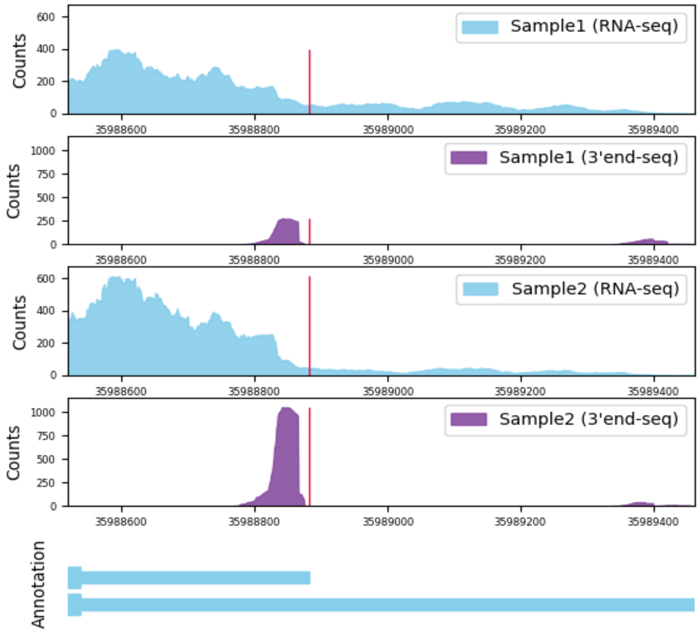
After executing this command above for a few seconds, **Make-plots.py** will ask the user to insert the region of interest in a specific format:
Chrom:GeneName:RegionStart-RegionEnd

Chrom **:**	Name of the chromosome
GeneName **:**	Name of the gene
RegionStart	Start position of the region

:	
Region End :	End position of the region

Example:
chr1:Tceb1:16641724-16643478

Make-Plots.py will generate a visual representation of the results shown for each of the regions entered. The plot will illustrate the most significant transcript cleavage site with a red vertical bar on top of RNA-seq read data (and 3'end-seq if available). If the input parameters have 3'end-seq information along with the RNA-seq, then it will generate plots for both cases (See figure below). It will also show the UTR truncation point (annotated and unannotated) at the bottom panel.


Counts
Sample1 (RNA-seq)
Counts
Sample1 (3'end-seq)
Counts
Sample2 (RNA-seq)
Counts
Sample2 (3'end-seq)
Annotation