**MANUAL For SMAtool 0.4.0**

**INTRODUCTION**

In order to facilitate the processing of eCLIP-seq data and study of RNA secondary structure’s roles in the protein-RNA interaction, we have developed a specialized software. It has four functions, including generation of secondary structure annotation reference, eCLIP-seq data processing, motif analysis and generation of figure (seq-motif distribution on structure-motif). Here, eCLIP-seq data processing procedure was performed to obtain significant enriched regions (peak) for downstream analysis. And secondary structure annotation reference for annotating eCLIP-seq enriched regions are generated based on RSF (RNA Structure Framework) processed results. Analysis of motif searching structure motifs and sequence motifs can contribute to investigate RNA binding protein binding bias for RNA. Finally, using generation of figure function can generate one figure for visualization.

**INSTALL**

1. Software requires Linux or Mac OS system environment with python 2.7 and terminal.
2. Python package dependencies: pandas, numpy, seaborn, matplotlib, itertools, bisect, scipy.interpolate.
3. Other tools dependencies: parallel in terminal, samtools and meme suit.
4. MEME software and forgi package are also required for the analysis. MEME software and forgi package are from websites: <http://meme-suite.org/doc/download.html?man_type=web>,<https://github.com/ViennaRNA/forgi/tree/v0.4.02>.
5. Download SMAtool from website: <https://github.com/QuKunLab/eCLIP>.
6. Raw data of eCLIP-seq must be in use with bam format of genome mapped (number of bam files should be three including two experimental files and one input control) and PARS or DMS data must be processed into dot-bracket format (dot-bracket example file shown in Section **1.1 Dot-bracket file example**) before the analysis.

**How to use**

All functions can be called with python script “SMAtool.py” and analysis must be step by step. Software details are descripted below.

* 1. **Generation of secondary structure annotation reference**

This function can be used for packaging annotation files with dot-bracket format processed by RSF (RNA Structure Framework), here is an example of command to execute it:

Usage: Python SMAtool.py <Options> [inputs] [outputs]

<Options>:

|  |
| --- |
| -t number of threads |
| --forgi path of forgi software in your system environment |

[inputs] must be a directory composed of dot-bracket files processed by RSF. [outputs] also should be assigned the path of a folder and output of this function includes annotated dot-bracket files and a file named “transcriptAnot.pickle” generated by packaging all annotated NAME.db files.

**Command Example:**

Python SMAtools.py -t 10 --forgi /Users/cai/soft/forgi -i ../name/dbs -o ../ref

**Dot-bracket file example**

>NM\_001353097

ACAUGCCAACAUGUGACGCGUGGAAUCUACGCCAGCCACAGCAGAGAGGUGGGGGUUGUGACCGGAGUUAGAUGCCAGCGCAGAAGGGACGUGAGCCCAGCCGAGGUACCGCUGAAGAGGAAUCAAUUUUGGAGGAAUUUUGUUGUCAGAGAAUAAAAGGAGGUUGUCCAUAAUUGACUUUAAGCAGCAAUCAGUAAAACAU

(((((....)))))...((((((...))))))((((((((((((((((((((((((((.(..((..(((...(((....))).....))).))..).)))))........(((((((((...((((((.((((..((((((.((............)).)))))).)))).))))))

**Output file example**

>NM\_001353097

ACAUGCCAACAUGUGACGCGUGGAAUCUACGCCAGCCACAGCAGAGAGGUGGGGGUUGUGACCGGAGUUAGAUGCCAGCGCAGAAGGGACGUGAGCCCAGCCGAGGUACCGCUGAAGAGGAAUCAAUUUUGGAGGAAUUUUGUUGUCAGAGAAUAAAAGGAGGUUGUCCAUAAUUGACUUUAAGCAGCAAUCAGUAAAACAU

(((((....)))))...((((((...))))))((((((((((((((((((((((((((.(..((..(((...(((....))).....))).))..).)))))........(((((((((...((((((.((((..((((((.((............)).)))))).)))).))))))

ssssshhhhsssssmmmsssssshhhssssssssssssssssssssssssssssssssisiissiisssiiissshhhhsssiiiiisssissiisisssssmmmmmmmmsssssssssmmmssssssissssiissssssisshhhhhhhhhhhh

* 1. **eCLIP-seq data processing**

This function can be used for processing eCLIP-seq data and we perform a specific method to get enough and significant enriched regions (peaks). Details are in the following example.

Usage: Python SMAtools.py <Options> [inputs] [outputs]

<Options>:

|  |
| --- |
| --b1 one experiment preprocessed bam file |
| --b2 another experiment preprocessed bam file |
| --bc control input bam file |
| --pv p-value threshold of significant enriched regions (peaks) (default = 0.01) |
| --rt relative value to define peak signal (default = 5) |
| --fc fold-change divided by control (default = 2) |
| --extend number to extend of central position (default = 20) |
| --ga transcript annotation reference |
| --anno PARS or DMS processed structure annotation reference (generated by function **1.1** ) |

[inputs] can be any path for this command because of input files existing, but [outputs] must be assigned the path of a folder.

**Command Example:**

Python SMAtools.py --b1 exp1.bam --b2 exp2.bam --bc ctr.bam --pv 0.05 --rt 5 --fc 2 –extend 20 --ga ../ref/hg19.txt --anno ../ref/transcript.pickle -i ../name -o ../name/eCLIP\_result

**Input files description:**

Input files are all bam files processed by mapping on genome. There are three files in total, two of which are experimental and one is input control. Besides, reference has two, one is transcript annotation reference (example file is attached in attachment) and one is structure annotation reference.

**Output files description:**

1. Only one output folder is named according to your definition.
2. NAME.rt is the normalized log file by calculating RT-Counts for each transcript site, where mock.rt is the input control.
3. bam.merged is a log file obtained by merging two experimental NAME.rt files.
4. bam.peak was obtained by peak calling for the experimental group.
5. peak.txt is the file obtained by comparing the peak value of the experimental group to the peak value of input control.
6. peak\_filtered.txt is obtained by filtering peak.txt file using threshold.
7. bam.struct is a file that annotates the secondary structure of peak.
8. bam.structAnot is the file that contains detailed secondary structure annotation for peak regions (default = 45). And this file is the input file in Motif analysis procedure. Contents of bam.structAnot descripted as below:

NM\_001012 377 AGUGGUACGAGUCCCACUAUGCGCUGCCCCUGGGCCGCAAGAAGG hhsssssssissssiississssiissshhhhsssssssssiiss 0.0 5.40772796038

**1.3 Motif analysis**

This function can be used for calling structure motifs and sequence motifs and it is taken by a two-step searching method. An example of command is descripted as followed:

Usage: Python SMAtools.py <Options> [inputs] [outputs]

<Options>:

|  |
| --- |
| --name protein’s name of definition |
| --alp alphabet of structure annotation (s, h, i, m). |
| --width structure motif length or width |
| --maxsite RBP’s binding sites or enriched regions on RNA |
| --mRNAref name of mRNA extracted from annotation file |

[inputs] must be bam.structAnot file processed by **section 1.2**,[outputs] also should be assigned a path of a folder.

**Command Example:**

Python SMAtools.py --name AARS --alp ../ref/dom\_structure.alphabet --width 25 --maxsite 1000 –mRNAref ../ref/mRNA.txt -i ../name/eCLIP\_result/bam.structAnot -o ../name/motif\_result

**Input files description:**

Input files must be bam.structAnot file by eCLIP-seq data processing. And mRNA reference can be extracted from transcript annotation reference (an example attached in attachment).

**Output files description:**

1. The output directory is determined by the command, and there are seven or eight folders in this directory (based on option “maxsite” in the command).
2. NAME\_NUMsites and NAME\_allsites were generated by performing sequence selection, secondary structure annotation selection around peaks, and values selection of peaks based on the number of binding sites for each protein on the transcript respectively. If the number of definition is greater than or equal to the number of peaks of a protein, only one folder was generated named NAME\_allsites.
3. NAME\_structMotif is a folder containing results of meme de-novo searching for structure motif using extracted secondary structure.
4. NAME\_structDataFrame is a folder containing results of detailed information for structure motif extracted from meme results.
5. NAME\_fimo is a folder containing results of fimo searching for structure motif for all peak regions.
6. NAME\_seq\_onStruct is a folder containing results of sequences extracted from all binding sites based on fimo results.
7. NAME\_seqMotif\_onStruct is a folder containing results of meme de-novo searching for sequence motif on structure motif using extracted sequences.
8. NAME\_seqDataFrame is a folder containing results of detailed information for sequence motif extracted from meme results.

**1.4 generation of figure**

This function can be used to generate one figure (seq-motif distribution on structure-motif) for visualization. Details of this function descripted as below:

Usage: Python SMAtool.py <Options> [inputs] [outputs]

<Options>:

|  |
| --- |
| --name2 protein’s name and it must be consistent with name defined in procedure “Motif analysis”. |
| --plotfile proteins's name and motif label for figure |

[inputs] must be consistent with “outputs” path defined in procedure “Motif analysis” and [outputs] should be assigned a path of a folder to save figure.

**Command Example:**

Python SMAtools.py --name2 AARS --plotfile ../name/plot\_file.txt –i ../name/motif\_result – o ../name/figure

**Input files description:**

Input files must be results generated by last step “Motif analysis”. And an example of option plotfile shown as below:

label

AARS\_1-1 AARS

AARS\_2-1 AARS

**Output files description:**

One figure shows seq-motif distribution on structure-motif and seq-motif is also shown in the bottom of figure.