**MANUAL For SMAtools 0.3.0**

**INTRODUCTION**

In order to facilitate the processing of eCLIP-seq data and to study the function of RNA secondary structure in the interaction between RNA binding protein and RNA, we have developed a specialized process of analysis for researchers. The analysis process has three main functions, including eCLIP-seq data analysis, generation of secondary structure annotation reference, motif analysis. Here, eCLIP-seq data analysis was performed to provide significant enriched regions for downstream analysis. And secondary structure annotation reference for annotating eCLIP-seq enriched regions are generated based on RSF (RNA Structure Framework) processing results. Analysis of motif searching structure motifs and sequence motifs can help scientists study RNA binding protein preference for RNA.

**INSTALL**

1. eCLIP-seq analysis require Linux or Mac OS system. Raw data of eCLIP-seq must be in use with bam format of genome mapped and PARS or DMS data must be processed into dot-bracket format before the analysis.
2. Python package dependencies pandas, numpy, itertools, bisect and tool dependencies parallel, samtools and meme suit .
3. Download SMAtools from website: <https://github.com/QuKunLab/eCLIP-pipe>.
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>.

\*Input bam file should have three files including two experimental files and one input control.

\*Control bam file can be used to filter experimental results to get significant peaks.

**How to use**

All functions can be called with python script “SMAtools.py” and you can do analysis step by step. The details are descripted below.

* 1. **eCLIP-seq analysis**

This function can be used for processing eCLIP-seq data and we perform a specific method to get enough and significant enriched regions. 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 |
| --rt reads threshold of experiments |
| --fc fold-change divided by control |
| --extend number to extend of central position |
| --ga genome annotation reference |
| --anno PARS or DMS processed structure annotation reference |

[inputs] can be any location for this command because of options 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 ../eCLIP/hg19.txt --anno ../eCLIP/transcript.pickle -i ../eCLIP -o ../eCLIP

* 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 SMAtools.py <Options> [inputs] [outputs]

<Options>:

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

[inputs] must be a directory of dot-bracket processed by RSF, [outputs] also should be assigned the path of a folder.

**Command Example:**

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

**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 results processed by section<1.1>, [outputs] also should be assigned the path of a folder.

**Command Example:**

Python SMAtools.py –name AARS –alp ../ref --width 25 –maxsite 4000 –mRNAref ../ref/mRNA.txt -i ../eCLIP/bam.structAnot -o ./motif

**Files description**

All input and output files of three functions were descripted as followed.

**2.1 eCLIP-seq analysis**

Input files are all bam files processed with mapping on genome. In our analysis, there are three files in total, two of which are experimental and one is input control.

Output folders and files:

1. Only one output folder is named according to your definition.
2. NAME.rt is the log file by calculating RT-Counts for each transcript site, where mock.rt is the control group.
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 after comparing the peak value of the experimental group with the peak value of the control group.
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 a file that makes a detailed secondary structure comment on peak. And this file will be used in Motif analysis.

**2.2 Generation of secondary structure annotation reference**

Input must be NAME.db format files. And this step was recommended using RNA Structure Framework(RSF).

Output is a pickle format file. And it was generated by packaging all annotated NAME.db files.

**2.3 Motif analysis**

Input files must be obtained by eCLIP-seq analysis. And annotation file must be pickle format obtained by step of generation of secondary structure annotation.

Output folders and files:

1. The output of the total directory is defined by its own name, and there are seven or eight folders in this directory.
2. NAME\_NUMsites and NAME\_allsites were generated by performing sequence selection, secondary structure selection around the peak, 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.