Manual for ATAC-pipe (1.0.0)

ATAC-pipe is composed by many deep analysis tools for ATAC-seq data besides reads alignment and quality control (QC), including cell-type classification, specific gene identification, transcription factors (TFs) footprint tracking, and regulatory network construction. Results provided by these procedures could help users to take full advantage of their experimental data, and understand the influence of chromatin accessibility on TF binding, transcription initiation, cell development and differentiation.

Installation

1. ATAC-pipe requires Linux or Mac OS system. Raw data should be produced by pair-end NGS and reported in FASTQ format.

2. Before the installation of ATAC-pipe, please install **Miniconda** to manage all needed software and dependencies. You can download **Miniconda** from https://conda.io/miniconda.html.

3. Download **ATAC-pipe.zip** from https://github.com/QuKunLab/ATAC-pipe. Unzipping this package you will see **condaEnv_atac-pipe.yml** located in its folder.

4. Build isolated environment for ATAC-pipe (This step will take several hours, depending on your internet and computer) :

```
conda env create -f condaEnv_atac-pipe.yml
```

5. Activate ATAC-pipe environment:

```
source activate atac-pipe
```

6. Download reference files and build reference indexes by command:

```
python refDown.py
```

*ATAC-pipe offers ‘hg19’ and ‘mm9’ as genome references. Other versions of reference genome are not applicable for current version.

**Pipeline runs as a command-line program with all commands in directory “<path>/ATAC-pipe/”.

***Every time you open a new terminal window to start an ATAC-pipe project, please active ATAC-pipe environment with “source activate atac-pipe”.

****Test datasets could be obtained from the following links:

http://galaxy.ustc.edu.cn:30803/zhangwen/ATAC-pipe/

Procedures for ATAC-pipe

In this chapter we only summarized every command and corresponding input parameters used by ATAC-pipe. More details for the step-by-step procedures are described in next chapter "**Step-by-step description**".

1. Easy-run for your convenience

Easy-run command for the pipeline is

```
python atac-pipe.py -i <input folder>
                    -o <output folder>
                    -r <genome reference>
                    --group <group information>
```

If you have less than 10 samples, and more than 10 CPU threads for each sample, you can use easy-run to executive step1~4 of ATAC-pipe at once (i.e. MappingQC, Merge, PeakCalling and SigAna steps in the table of Section 2). Here we list required parameters for easy-run command:

`-i` REQUIRED parameter. DIRECTORY of input pair-end fastq files. Files must be named as NAME_1.fq or NAME_2.fq and '.' is not allowed in NAME.

`-r` REQUIRED parameter. REFERENCE GENOME. ONLY can be 'hg19' or 'mm9'.

`-o` REQUIRED parameter. DIRECTORY of output files, where folder 'Mapping', 'QC', 'Group', 'Peak', 'Sig' will be constructed.

`--group` REQUIRED parameter. FILE contains group information. Its first column should be sample names, and the second column should represent corresponding group names. All terms should be separated by TAB.

Note: All optional parameters in step1~4 are also applicable in easy-run. Other required parameters in step1~4 are not available in easy-run.

2. Functions

If you want to run ATAC-pipe step-by-step, please follow the basic format to submit each functional command as

```
python atac-pipe.py <Function> <Parameters | options>
```

All functions should be run in the order listed as following table. More details about each command, corresponding parameters and output files can be seen in next Chapter "**Step-by-step description**".

Function	Parameters & options	Description
`--MappingQC`	`-i` <input folder> `-o` <output folder> `-t` <number of CPUs> `-c` <cutoff>	1. Raw reads alignment 2. File format conversion 3. Quality control
`--Merge`	`--bam` <input bam-files folder> `-o` <output folder> `-r` <genome reference> `--group` <group information>	Sample grouping
`--PeakCalling`	`--bed` <input bed-files folder> `-o` <output folder> `-r` <genome reference> `--group` <group information> `--p1` <P-value> `--q1` <Q-value> `--f1` <fold change> `--pipeup` <counts on peak> `-u` <extend peak or not> `-w` <peak width> `--project` <project name>	1. Peak Calling for each merged group with MACS2 2. Reads counting for each sample
`--SigAna`	`--count` <input count file> `--peak` <input peak list> `--group` <group information> `-o` <output folder> `--p2` <P-value>	Significant differential accessible regions for pairwise samples comparison by DESeq

	`--q2` <Q-value> `--f2` <fold change>	
`--SearchMotif`	`--cluster` <input cluster list> `-r` <genome reference> `-o` <output folder> `--bg` <input background list> `--size` <batch size of peaks>	Motif searching and enrichment analysis by HOMER
`--Footprint`	`--motif` <TF-motif name> `--peak` <peak-list> `-o` <output dir> `--perbase` <output bed file>	Footprint and V-plot illustration for a specific motif
`--Network`	*described in Section 3	Regulatory network construction between different sample categories

3. Regulatory network

The construction of TF regulatory network is not contained in Easy-run version of the pipeline, and msCentipede is adopted for the estimation of posterior probabilities. Users have to run following steps to build TF regulatory network between two categories of samples:

(1) `python setup.py build_ext –inplace`

(2) `python step1_prepare_data_files.py -r <ref> --tf <TFs file>`

(3) `python step2_run_centipede.py <ref> <CPUs> <bam file 1>`

`<output folder> <TF name>`

**You should run this step for every motif of your interest on each sample.

(4) `python step3_build_clustermap.py <CPUs> <ref>`

`<output folder> <TFs file>`

`<bam file 1> <bam file 2>`

More details about parameters and result for this step are in Section 6 of next Chapter "**Step-by-step description**".

Step-by-step description

1. MappingQC

MappingQC is the first step for ATAC-pipe, which make reads alignment and quality control with command as:

```
Python atac-pipe.py --MappingQC <parameters | options>
```

Necessary and optional parameters for MappingQC:

`-i` REQUIRED parameter. DIRECTORY of input pair-end fastq files. Files must be named as NAME_1.fq and/or NAME_2.fq, and '.', '_' are not allowed in NAME.

`-r` REQUIRED parameter. REFERENCE GENOME. ONLY 'hg19' or 'mm9' is alternative.

`-o` REQUIRED parameter. DIRECTORY of output files, under which folder '**Maping**' and '**QC**' will be built by the pipeline.

`-c` Optional parameter. INTEGER. If you set "-c 500", then only the first 500 base of sequence read will be used for alignment. Default = 1000.

`-t` Optional parameter. INTEGER. Total CPU threads required. All threads will be allocated equally to each sample. Default = 4.

`--aq` Optional parameter. INTEGER. Query length for adapter trimming, default = 20.

`-a` Optional parameter. STRING. Adapter sequence. Default = 'CTGTCTCTTATACACATCTGACGCTGCCGACGA'

Output folders and files:

(1) Files in folder **"<outputdir>/Mapping/"**

NAME.pe.q10.sort.rmdup.bam is a BAM format file which removes all chrM and duplicated reads. If you want to merge samples by groups, this file is recommended to use.

NAME.pe.q10.sort.rmdup.shift.bed is a BED format file which contains location information for each read. The second and third column are start site and end site of each read after shifting read +4/-5 bp (depending on +/- strand) and extending it to 50bp around the cleavage sites. The central base of read representing the Tn5 binding position. The fifth column of this file is the length of fragment corresponding to each read. This file will be used for peak calling.

NAME.pe.q10.sort.rmdup.shift.norm.bw is a BigWig format file, which has been normalized by sequencing depth. This file can be loaded directly to UCSC genome browser.

NAME.pe.q10.sort.rmdup.shift.per1base.bed is a BED format file, which is similar to shift.bed file, except for adjusting Tn5 binding base as the start site and next base as the end site.

NAME.pe.q10.sort.rmdup.shift.per1base.bedGraph is a BedGraph format file, which contains genome-wide counts of Tn5 binding events at each base pair. This file will be used to depict footprint of TF motifs.

(2) Files in folder **"<outputdir>/QC/"**

NAME_4kb_TSSenrichment.pdf is a PDF format file, which illustrate reads enrichment near all transcript start site (TSS). In this figure, x-axis is the distance from read to TSS center, and y-axis represents relative count of reads at corresponding distance. Each point represent a read, and the red line is the average curve. According to our experience, a good sample shows average enrichment score higher than 5.5 at TSS.

NAME_Fragmentdistribution.pdf is a PDF format file, which depicts the distribution of fragment length. In this figure, x-axis represents fragment length and y-axis is the count of fragments with corresponding length. User should see three obvious peak summits around 50 bp, 200 bp and 400 bp, respectively, if the sample data has good quality.

NAME.qctable is a TABLE file which contains sample's quality information. All terms in this file are described as:

Sample	Sample name
TotalRawReads	Total count of all raw reads
OverallAlignmentRate %	Rate of reads mapped on genome
FinalMappdReads	All mapped reads, except for reads mapped on chrM, reads in blacklist, reads with low MAPQ, and duplicated reads.
FinalMapped%	Rate of final mapped reads
chrM%	Rate of reads mapped on chrM
BlackListReads%	Rate of reads in blacklist
MAPQFiltered%	Rate of low MAPQ reads
Duplicate%	Rate of duplicated reads

2. Merge

As the second step of ATAC-pipe, process Merge groups all samples by command:

```
Python atac-pipe.py --Merge <parameters | options>
```

Necessary parameters for Merge:

`--bam` REQUIRED parameter. DIRECTORY of input bam files. Files must be named as NAME.pe.q10.sort.rmdup.bam and '.' is not allowed in NAME.

`-r` REQUIRED parameter. REFERENCE GENOME. ONLY can be 'hg19' or 'mm9'.

`-o` REQUIRED parameter. DIRECTORY of output files, in which folder 'Group' will be constructed.

`--group` REQUIRED parameter. FILE contains group information. Its first column should be sample names, and the second column should represent corresponding group names. All terms should be separated by TAB.

Output files:

Files in folder **"<outputdir>/Group"**:

GroupName.pe.q10.sort.rmdup.shift.bw is a BigWig format file, similar to NAME.pe.q10.sort.rmdup.shift.bw, but contains all samples that belong to one GroupName. This file can be loaded directly to UCSC genome browser.

GroupName.pe.q10.sort.rmdup.shift.per1base.bed is a BED format file, which is similar to NAME.pe.q10.sort.rmdup.shift.per1base.bed, but contains all samples that belong to one GroupName. This file will be used to illustrate V-plot for TF-motifs.

GroupName.pe.q10.sort.rmdup.shift.per1base.bedGraph is a BedGraph format file, which is similar to NAME.pe.q10.sort.rmdup.shift.per1base.bedGraph, but contains all samples that belong to one GroupName. This file will be used to plot footprint for motifs.

GroupName.pe.q10.sort.rmdup.shift.per1base.norm.bw is a BigWig format file, which is similar to NAME.pe.q10.sort.rmdup.shift.per1base.norm.bw, but contains all samples that belong to one GroupName. This file can be loaded directly to UCSC genome browser.

GroupName_4kb_TSSenrichment.pdf and GroupName_Fragmentdistribution.pdf are PDF format files, which are similar to NAME_4kb_TSSenrichment.pdf and NAME_Fragmentdistribution.pdf respectivley, but contain all samples that belong to one GroupName.

GroupName_ReadDenstiyTSS.png is a PNG format file, which depicts fragment's density near all TSSs. In lower panel, column is fragment density sorted by length distribution. Negative number means the fragment header is on the left side of the TSS, while positive number means the fragment header on the right side of the TSS. Each row is the fragment density profile of each TSS and ranked by total number. Figure in upper panel is similar to GroupName_Fragmentdistribution.pdf, but only counts fragments near TSSs.

3. PeakCalling

The third step of ATAC-pipe, i.e. PeakCalling, can be used with command:

```
Python atac-pipe.py --PeakCalling <parameters | options>
```

Necessary and optional parameters for PeakCalling:

`--bed` REQUIRED parameter. DIRECTORY of input bed files. Files must be named as NAME.pe.q10.sort.rmdup.bed and '.' is not allowed in NAME.

`-r` REQUIRED parameter. REFERENCE GENOME. ONLY can be 'hg19' or 'mm9'.

`-o` REQUIRED parameter. DIRECTORY of output files, where folder 'Peak' will be constructed.

`--group` REQUIRED parameter. FILE contains group information. Its first column should be sample names, and the second column should represent corresponding group names. All terms should be separated by TAB.

`--p1` OPTIONAL parameter. INTEGER. Parameter for high quality peaks filtering. Peaks with $|\log_{10}(\text{P-value})| > p1$ will be selected. Default = 2

`--q1` OPTIONAL parameter. INTEGER. Parameter for high quality peaks filtering. Peaks with $|\log_{10}(\text{Q-value})| > q1$ will be selected. Default = 5

`--f1` OPTIONAL parameter. INTEGER. Parameter for high quality peaks filtering. Peaks with fold enrichment > f1 will be selected. Default = 1

`--pipeup` OPTIONAL parameter. INTEGER. Parameter for high quality peaks filtering. Peaks with mapped reads > pipeup will be selected. Default = 10.

`-w` OPTIONAL parameter. INTEGER. Parameter for peak adjust. If you set “-w 500”, every peak will be extend to 500 bp length by taking peak summit as center. Default = 500.

`-u` OPTIONAL parameter. INDICATOR. If you run PeakCalling with “-u”, peak will not be extend.

`--project` OPTIONAL parameter. STRING. Name for the project and '.', '_' are not allowed in Name. Default = “atac”

Output files:

Files in folder **“<outputdir>/Peak”**:

ProjectName.atac.merged.peak.list is a BED format file which lists all open regions/peaks. In the fourth column there are peak IDs, and the 5th column is the number indicate how many samples have signal on one peak.

ProjectName.count is a N×M data matrix where N is the number of merged peaks, M is the number of samples, and element $D_{i,j}$ indicates the original intensity of peak i (i=1~N) in sample j (j=1~M).

4. SigAna

SigAna is the fourth step of ATAC-pipe. Its command is:

```
Python atac-pipe.py --SigAna <parameters | options>
```

Necessary and optional parameters for SigAna:

`--count` REQUIRED parameter. FILE. A N×M data matrix where N indicates the number of merged peaks, M indicates the number of samples, and value $D_{i,j}$ indicates the original intensity of peak i (i=1~N) in sample j (j=1~M).

`--group` REQUIRED parameter. FILE contains group information. Its first column should be sample names, and the second column should represent corresponding group names. All terms should be separated by TAB.

`--peak` REQUIRED parameter. FILE. Output file of PeakCalling step, which contains peak position for each peak ID.

`-o` REQUIRED parameter. DIRECTORY of output files, in which folder

'Sig' will be constructed.

`--p2` OPTIONAL parameter. FLOAT. Parameter for significant peaks filtering. Peaks with P-value > p2 will be selected. Default = 0.01

`--q2` OPTIONAL parameter. FLOAT. Parameter for significant peaks filtering. Peaks with Q-value < q2 will be selected. Default = 0.01

`--f2` OPTIONAL parameter. FLOAT. Parameter for significant peaks filtering. Peaks with $|\log_2(\text{Fold Change})| > f1$ will be selected. Default = 0.05

Output files:

Files in folder **"<outputdir>/Sig"**:

ProjectName.sigdata is a MATRIX which contains significant counts. This file can be applied directly to Cluster3.0 for hierarchy cluster.

Guideline to use Cluster3.0 & TreeView

We recommend users to adopt cluster3.0 and TreeView for hierarchy cluster with SIGDATA FILE produced in step4 or easy-run. Here is a simple instruction for Cluster3.0 and TreeView:

1. Launch a new Cluster3.0 window.

2. In main menu, click "File", then click "Open Data", select SIGDATA FILE, click 'Open' to complete file upload.

3. In main panel, click 'FilterData', uncheck all boxes. Then click "ApplyFilter", it will show "No. passed out of No.". Then click "Accept".

4. Click "Adjust Data", select check boxes before "Center genes" and "Mean", make sure that all other check boxes are unselected. Click "Apply".

5. Click "Hierarchical", select the check boxes before "Cluster" in 'Genes' and "Arrays", select similarity metric in drop-down menu (Euclidean distance is

recommended). Then select a clustering method (Average linkage is recommended).

6. Wait for cluster to finish. You will get files end with suffixes cdt, atr, gtr, jtv in the same directory with SIGDATA FILE. This file can be used for visualization with TreeView.

7. Launch a new TreeView window.

8. In main menu, click "File", then click "OpenData", select SIGDATAFILE, click 'Open' to complete file upload.

9. Click "Setting", "PixelSetting" to set display options. Usually, we suggest user to choose 'Fill' for 'Global' and 'Zoom' options, '1 to 3' for 'Contrast' option. 'Log(base2)' is not recommended to use.

10. To save peak list of specific pattern, select pattern of your interest, click 'Export', 'SaveList', rename output file and click "Save".

11. To save figures, click 'Export'. There are several options for export format, choose one and click 'Save'.

5. SearchMotif

SearchMotif is the fifth step of ATAC-pipe, with command as:

```
Python atac-pipe.py --SearchMotif <parameters | options>
```

Necessary and optional parameters for SearchMotif:

`-r` REQUIRED parameter. REFERENCE GENOME. ONLY can be 'hg19' or 'mm9'.

`-o` REQUIRED parameter. DIRECTORY of output files, where folder 'Motif' will be constructed.

`--peak` REQUIRED parameter. FILE. Output file of PeakCalling which

contains peak position for each peak ID.

`--cluster` REQUIRED parameter. FILE. Output file of TreeView which is a list of peak IDs of a specific pattern.

`--bg` OPTIONAL parameter. FILE. Output file of TreeView which acts as background peak list here.

Output files:

Files in folder **"<outputdir>/Motif"**:

knownmotif.html is a HTML file which contains known motifs searched from the peak list.

homermotif.html is a HTML file which contains de nova motifs searched from the peak list.

6. Footprint

Footprint is the sixth step of ATAC-pipe, with command as:

```
Python atac-pipe.py --Footprint <parameters | options>
```

Necessary and optional parameters for Footprint:

`--motif` REQUIRED parameter. STRING (in lowercases). Motif name for footprint and nucleosome occupancy depicting, such as "ctcf".

`--peak` REQUIRED parameter. FILE. Output file of PeakCalling which contains peak position for each peak ID.

`-r` REQUIRED parameter. REFERENCE GENOME. ONLY can be 'hg19' or 'mm9'.

`-o` REQUIRED parameter. DIRECTORY of output files, in which folder

'Footprint' will be constructed.

`--perbase` REQUIRED parameter. FILE. A perbase file in BED format. Output results of MappingQC or Merge.

Output files:

Files in folder **"<outputdir>/Footprint"**:

Peak_Motif_BED.footprint.png is a PNG file which depicts TF's footprint around Motifs from the Peak list in the perbase BED file.

Peak.include.Motif.in.BED.png is a PNG file which depicts nucleosome occupancy around Motifs from the Peak list in the perbase BED file.

7. Regulatory network

To build TF regulatory network between two different types of samples, you need to run following command one by one:

(1) `python setup.py build_ext –inplace`

(2) `python step1_prepare_data_files.py -r <ref> --tf <TFs file>`

(3) `python step2_run_centipede.py <ref> <CPUs> <bam file 1>`
`<output folder> <TF name>`

** You should run this step for every motif of your interest on each sample.

(4) `python step3_build_clustermap.py <CPUs> <ref>`
`<output folder> <TFs file>`
`<bam file 1> <bam file 2>`

Here we list the meanings for all upon parameters:

`<ref>` REQUIRED parameter. REFERENCE GENOME. ONLY can be 'hg19' or 'mm9'.

`<TFs file>` REQUIRED parameter. FILE. List for motif names (in lowercases).

`<CPUs>` REQUIRED parameter. INTEGER. Number of CPU Threads used by the pipeline.

`<output folder>` REQUIRED parameter. DIRECTORY. Output folder.

`<bam file 1>` REQUIRED parameter. FILE. Perbase file in BAM format, output file of type 1 sample from MappingQC or Merge step.

`<bam file 2>` REQUIRED parameter. FILE. Perbase file in BAM format, output file of type 2 sample from MappingQC or Merge step.

Output files:

<bam name 1>_vs_<bam name 2>.TF_network.pdf is a PDF file with a heatmap figure to illustrate TF regulatory correlation network between two types of samples.