Quick Start for CAM

CAM is a quality control (QC) pipeline for MNase-seq data. By applying this pipeline, CAM uses either raw sequencing file or aligned file as an input (supporting both paired end and single end data; see our testing data and the Manual section for more information) and provides multiple informative QC measurements and nucleosome organization profiles on potentially functionally related regions for a given MNase-seq dataset. CAM also includes 268 historical MNase-seq datasets from human and mouse as a reference atlas for unbiased assessment.

Here, we provide an example to get you easily start in 3 steps on a Linux/MacOS system with only Python and R installed. You can use the default mode to run CAM with options specific to your data (see the Manual section for detailed usage).

STEP1.Install the pipeline

1. Make sure you have python(version = 2.7) and R(version >= 2.14.1) on linux or MacOS environment. Type "python" or "R" in Terminal to check.

2. Get CAM from the link below (size of each package is about 100M, because we include many annotation files in the packages for users convenience)

CAM.1.2.linux.x86_64

CAM.1.2.macOSX.x86_64

CAM.1.2

3. Install CAM on your server/computer (Please contact the administrator of that machine if you want their help to install in the public environment)

```
$ unzip CAM.1.2.linux.x86_64.zip   # use linux version as example
$ cd CAM.1.2.linux.x86_64    # find your CAM.1.2.linux.x86_64 folder and change working
directory to it
```

for the root user

```
$ sudo python setup.py install
```

if you are not a root user, you can install CAM at specific locations which you have write permission

```
$ python setup.py install --prefix /home/CAM    # here you can replace “/home/CAM" with any location you want
$ export PATH=/home/CAM/bin:$PATH    # setup PATH, so that system knows where to find executable files
$ export PYTHONPATH=/home/CAM/lib/python2.7/site-packages:$PYTHONPATH    # setup PYTHONPATH, so that CAM knows where to import modules
```

Does exporting the environmental variables every time bother you? In linux, Users can add these command lines to .bashrc (or .bash_profile) file to automatically export PATH and PYTHONPATH. See "DIY template configure file to save your specific parameters" in the manual section.

NOTE: To install CAM on MacOS, user must download and install Command Line Tools beforehand

Type:

```
$ CAM.py --help
```

If you see help manual, you have successfully installed CAM.

STEP2.Prepare the annotation

Obtain a genome sequence file (e.g., hg19.2bit or hg19.fa) according to the species of your sample.
CAM supports human (hg38 and hg19) and mouse (mm10 and mm9) genome versions, you can download the genome sequence file in twobit format from the link below:

hg38.2bit (size: 797M)

hg19.2bit (size: 778M)

mm10.2bit (size: 682M)

mm9.2bit (size: 680M)

The genome sequence file is for --fa parameter of CAM. Both .fa and .2bit file (e.g. hg19.fa and hg19.2bit) can be used here. For example, if you already have hg19.fa, you can just use hg19.fa and don't need to download hg19.2bit here. See the description in the next step. If you can download genome sequence file from NONE of above links, you can also download genome sequence file from UCSC genome browser http://genome.ucsc.edu/cgi-bin/hgTables, see the Manual section.
NOTE: This is the ONLY required annotation file for default CAM, you don't need to input any other file to use full functions of CAM.

STEP3.Run CAM

Before you running the progream, CAM checks your computer for pdflatex. If you previously installed pdflatex, CAM will generate a summary QC report in addition to QC and analysis results (see the Manual section for the installation of pdflatex).

Make sure your server has enough space/memory for CAM. MNase-seq sample with 500M reads will occupy about 70G space and the memory usage also related to the sequencing coverage of MNase-seq data.

Now you can run the CAM pipeline to generate QC and analysis results of your MNase-seq data using FASTQ, SAM or BED files as the input.
Here we provide an example of our simple mode on published MNase-seq data (GSM907784) and display the CAM output in the following panel.

Below is an example command for CAM:

```
$ CAM.py simple -a GSM907784_1.fastq -b GSM907784_2.fastq -n GSM907784 -t PE -s hg19 --fa
/home/data/hg19.fa -c /home/data/CTCF_motif_hg19.bed
```

Brief description of major parameters; see the Manual section for more information

if your data is a paired end data, FASTQ format, you need -a data_1.fastq -b data_2.fastq, otherwise, you only need -a data.fastq/data.sam/data.bed (for paired end data in SAM/BED format, only need -a)
-a GSM907784_1.fastq: Sequencing data from a MNase-seq experiment: accept raw sequencing data (FASTQ) and aligned data (SAM/BED)

-b Second part of the paired end fastq file, only works for paired end data and FASTQ format as the input

-t Sequencing type, choose from PE (paired end) and SE (single end)

-n Name of your output results and output directory, note that no "/" should be appeared here

-g /home/user/annotation/mm10_refgenes.txt: Absolute path of the genome annotation file

--fa /home/data/hg19.fa: Absolute path of genome sequence file. Both .fa and .2bit file can be used for this parameter (e.g. hg19.fa and hg19.2bit). It's exactly what you downloaded in step2. The genome sequence file should be corresponding to the -s (species) parameter.

-s hg19: Species (genome version) to specify the genome length and gene annotation. Should be corresponded to the --fa parameter.

--mapindex [optional] /home/user/bowtie_index/hg19, user can specify bowtie mapping index by this parameter to skip bowtie-build step, which is time consuming. See "Prepare bowtie mapping index files" in the manual section.

-c [optional] /home/data/CTCF_motif_hg19.bed: Custom regions, user can generate nucleosome profile on other potential target regions by specify this parameter, the example custom region file can be downloaded in Download section. This function will be turned off if the -c parameter is not specified. see Download section for the example of custom region.

-f [optional] whether to remove the existing output folder and restart the program.

--task [optional] choose from 1 to 4, defining to process which procedures. ‘1’ for preprocessing data, ‘2’ for detecting nucleosome arrays, ‘3’ for quality control and ‘4’ for all procedures including the summary. The program will check the intermediate files automatically, and complete interrupted procedures from the last running if the parameter ‘-f’ is not True.

Example of custom region file (CTCF motif sites in hg19): CTCF_motif_hg19.bed (zip compressed)

After you finish CAM, find your result in the outname/summary/ folder.