

Day 9 Worksheet – MACS

Author: Jessica Westfall (adapted from Joe Cardiello SR2019):

MACS <https://github.com/macs3-project/MACS>

Introduction: To study DNA enrichment assays such as ChIP-seq and ATAC-seq, we are introducing the analysis method, Model-based Analysis of ChIP-Seq (MACS). This method enables us to identify transcription factor binding sites and significant DNA read coverage through a combination of gene orientation and sequencing tag position.

- ! Note: The directory and username used in the screenshot will be for my working directory and username and will be different than yours.

Make working directories

Similar to previous worksheets, make the necessary working directories for running MACS. Repeat the same process, but this time we will make a directory for trimmomatic.

1. Use command `pwd` to determine what directory you are in and if necessary, `cd` to the directory that you want to place your new macs directory in.
2. Make a new directory using the `mkdir` command. Use command `ls -lsh` to confirm the folders are present.

```
[~bash-4.2$ mkdir macs2 macs2/output macs2/scripts
```

MACS

3. Copy (`rsync` or `cp`) the `d9_macs.sbatch` from the sample script folder into your script directory that you made in the previous exercise. Recall to copy the script, the command syntax is `rsync <input><output>` or `cp`.
4. Edit the sbatch script by using `vim <sbatch>` to open a text editor on your sbatch script. Type `i` to toggle into edit/insert mode. Similar to the previous exercise you will need to change the job name, user email, and the standard output and error log directories. Change the `-job-name=<JOB_NAME>` to a name related to the job you will be running, for example 'trim_qc'. Additionally you will want to change the `-mail-user=<YOUR_EMAIL>` to your email, as well as the path to your efiles directory for the standard output (`--output`) and error log (`--error`). The `%x` will be replaced by your `-job-name` and the `%j` will be replaced by the job id that will be assigned by the job manager when you run your sbatch script.

```
#!/bin/bash
#SBATCH --job-name=macs2                # Job name
#SBATCH --mail-type=ALL                 # Mail events (NONE, BEG
IN, END, FAIL, ALL)
#SBATCH --mail-user=<USER@EMAIL.COM>    # Where to send mail
#SBATCH --nodes=1                      # Numbers of nodes
#SBATCH --ntasks=1                    # Number of CPU (tasks)
#SBATCH --time=00:30:00                # Time limit hrs:min:sec
#SBATCH --partition=compute            # Partition/queue reques
ted on server
#SBATCH --mem=2gb                      # Memory limit
#SBATCH --output=/path/to/eofiles/%x_%j.out
#SBATCH --error=/path/to/eofiles/%x_%j.err
```

5. **module load**. To run MACS, we will need to load python since MACS is dependent on it. In addition we will want to load bedtools which we will use later to remove Blacklist regions.

```
##### LOAD MODULES #####
module load python/2.7.14
module load python/2.7.14/MACS/2.1.1
module load bedtools/2.25.0
```

6. **Set variable**. Assigning variables will make your scripts easier to read. In addition, this makes it easier to reference to a given path and utilize it in your scripts.

For the **INDIR=** change the path to the bam files directory. We will be using bam file from ChIP-seq data that used a specific transcription factor. For the **OUTDIR=**, point to the appropriate output file directories for our MACS output files. You can use the command **mkdir -p** just in case for my output directories if you want to ensure that the output directory exist.

In addition, I have a path to **BLACKLIST** directory. These are regions that have been identified as having unstructured or high signal in Nextgen sequencing experiment independent of the cell line or experiment. Removing these will clean up our genomic data for improved quality measurement. ENCODE has a defined list. The list we are using comes from the following reference: Amemiya HM, Kundaje A, Boyle AP. The ENCODE blacklist: identification of problematic regions of the genome. Sci Rep. 2019 Dec; 9(1) 9354 DOI: [10.1038/s41598-019-45839-z](https://doi.org/10.1038/s41598-019-45839-z)

Lastly, I am using the variable **FILENAME** so that I can quickly interchange different files for analysis and only have to change the variable rather than go through my script to change instances of the file.

```
##### SET VARIABLES #####
#INDIR is where my bams file are stored. OUTDIR is where I want my output from
running MACS to go
INDIR='/path/to/bam/file'
OUTDIR='/path/to/output/diretory'

BLACKLIST='/scratch/Shares/dowell/sread/data_files/day9/blacklist_hg38/problematic_regions_hg38.bed'

FILENAME='BACH1'
```

7. To run the MACS program, we have many different subcommand options. Depending on your experiment, you will want to change the subcommands to fit your requirement.

Usage

```
macs2 [-h] [--version]
      {callpeak,bdgpeakcall,bdgbroadcall,bdgcmp,bdgopt,cmbreps,bdgdifff,filterdup,predictd,pileup}
```

Example for regular peak calling: `macs2 callpeak -t ChIP.bam -c Control.bam -f BAM -g hs -n test -B -q 0.01`

Example for broad peak calling: `macs2 callpeak -t ChIP.bam -c Control.bam --broad -g hs --broad-cutoff 0.1`

There are twelve functions available in MAC2S serving as sub-commands.

Reference: <https://pypi.org/project/MACS2/>

For today's worksheet, we will be showing an example where we utilized an input control with your experiment.

-t / --treatment <filename> is your experimental file. The file can be in any supported format (see `--format` for options). If you have more than one alignment file, you can specify them and MACS will pool all the files together.

-c / --control <filename> is your genomic input/control file.

-n / --name <NAME> is the name string of your experiment. The string **NAME** will be used by MACS to create output files.

-B/ --BDG flag to tell MACS to store the fragment fileup, control lambda in bedGraph files.

-g / --gsize <GENOME> is the parameter to assign the mappable genome size. We will be using `hs` which is the recommended human genome size of `2.7e9`.

-q / --qvalue <VALUE> is the cutoff to call significant regions. The default is `0.05`. If you want to use a p-value cutoff, you can specify **-p** instead of **-q**.

Note that there are many other options than the ones that we are implementing here.

```
echo macs2
date
date

#### Call peaks with controls
macs2 callpeak \
  -c ${INDIR}/${FILENAME}.input.chr21.sorted.bam \
  -t ${INDIR}/${FILENAME}.chr21.sorted.bam \
  --outdir ${OUTDIR} \
  -n ${FILENAME} \
  -g hs \
  -B \
  -q 0.01 \
```

If you wanted to run to get Broad peaks you will want to use the flag **-broad**

8. Removing Blacklist regions via **bedtools intersect**. After we call our peaks, to clean up the data we will remove the **BLACKLIST** regions that can be problematic. These regions contain repetitive regions across the genome and almost always are enriched in ChIP-seq data.

To run **bedtools intersect**, specify **-a** as the file to be filter which is your broadpeak output file. The **-a** file will be compared against **-b** file which is the blacklist regions. The **-v** parameter will throw out the regions in your peak files that have an overlap with the blacklist regions in **-b**. **>** is to specify the output directory and output file name.

```
echo removing blacklist regions
date
date

bedtools intersect \
  -a ${OUTDIR}/${FILENAME}_peaks.broadPeak \
  -b ${BLACKLIST} \
  -v \
  > ${OUTDIR}/${FILENAME}_peaks_clean.broadPeak

echo macs2 done
date
date
```

9. **sbatch** and run your script. If you want to change the filename input you can do so as show in step 6.

MEME and TOMTOM

MEME suite is useful for motif-based analysis. In the next session, we will briefly demonstrate how you can use MEME for motif discovery and TOMTOM to compare motifs.

1. Copy (**rsync** or **cp**) the **d9_meme.sbatch** from the sample script folder into your script directory that you made in the previous exercise. Recall to copy the script, the command syntax is **rsync <input><output>** or **cp**.

2. MEME suite takes in a fasta file as input. Our MACS peak output is in a bed file format. We will use **bedtool getfasta** and a reference genome .fa file to convert our peaks coordinate into a fasta format. The first thing we will do in our script is to load the appropriate modules.

```
##### LOAD MODULES #####  
  
module load bedtools/2.25.0  
module load meme/5.1.1
```

3. Set your in and out directory as we have in the previous exercise. Here your **INDIR** is the path to your macs peak output files. The **OUTDIR** will be for the output of the fasta file and the MEME and TOMTOM output files. Additionally, we will want a reference fasta file denoted below as **hg38.fa**

```
##### SET VARIABLES #####  
  
#INDIR is where your macs output peak files are stored. OUTDIR is where I  
want my output from running meme and tomtom  
INDIR='/path/to/macs/peak/files'  
OUTDIR='/path/to/output'  
mkdir -p ${OUTDIR}  
  
HG38_FASTA='/scratch/Shares/dowell/genomes/hg38/hg38.fa'  
  
FILENAME='BACH1'
```

4. We will use **bedtools getfasta** to convert the peaks to a fasta file to feed into MEME. The command is **bedtools getfasta [OPTIONS] -fi <input FASTA> -bed <BED/GFF/VCF>**

```
#### Get fasta of peak files  
echo convert peaks call to fasta format  
date  
date  
  
bedtools getfasta \  
-fi ${HG38_FASTA} \  
-bed ${OUTDIR}/${FILENAME}_peaks_clean.narrowPeak \  
-fo ${OUTDIR}/${FILENAME}_peaks_clean.fasta
```

5. For running MEME and TOMTOM, you can use command line which I have included a sample script commented out. MEME suite also has a web interface which is what we will use. Rsync your fasta file onto your local drive
6. Upload your fasta file to MEME (<https://meme-suite.org/meme/tools/meme>) and submit.

Data Submission Form

Perform motif discovery on DNA, RNA, protein or custom alphabet datasets.

Select the motif discovery mode [?](#)

Classic mode Discriminative mode Differential Enrichment mode **NEW**

Select the sequence alphabet

Use sequences with a standard alphabet or specify a custom alphabet. [?](#)

DNA, RNA or Protein Custom No file chosen

Input the primary sequences

Enter sequences in which you want to find motifs. [?](#)

BACH1.fasta [?](#)

Select the site distribution

How do you expect motif sites to be distributed in sequences? [?](#)

Select the number of motifs

How many motifs should MEME find? [?](#)

Input job details

(Optional) Enter your email address. [?](#)

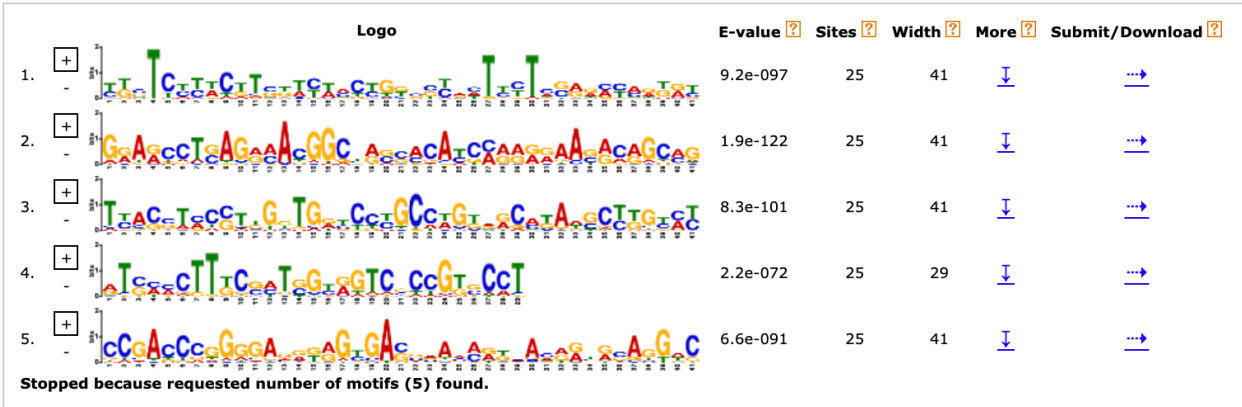
(Optional) Enter a job description. [?](#)

Advanced options

Note: if the combined form inputs exceed 80MB the job will be rejected.

7. MEME will return an output file for you. Click on **MEME HTML output**. The output will give you information on the motifs that were discovered along with other information such as the E-value.

DISCOVERED MOTIFS



8. Push your MEME output to [TOMTOM](#) by clicking on the [→](#) under Submit/Download which will open up a new window with available programs. You just have to Start Search to run TOMTOM.

Data Submission Form

Search one or more motifs against a motif database.

Input query motifs

Enter the motif(s) to compare to known motifs. [?](#)

Submitted motifs [▼](#) [DNA](#) [?](#)

Select target motifs

Select a [motif database](#) or provide motifs to compare with. [?](#)

Eukaryote DNA [▼](#) [DNA](#) [?](#)

Vertebrates (In vivo and in silico) [▼](#) [?](#)

Allow alphabet expansion [?](#)

Run immediately

Search with one motif (faster queue) [?](#)

Input job details

(Optional) Enter a job description. [?](#)

▶ Advanced options

Note: if the combined form inputs exceed 80MB the job will be rejected.

Start Search

Clear Input

