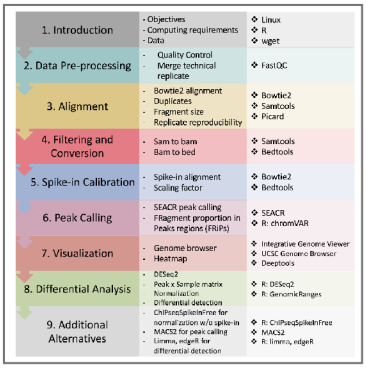


VERSION 1

FEB 27, 2024

OPEN ACCESS

External link:
https://yezhengstat.github.io/CUTTag_tutorial/

Protocol Citation: Ye Zheng, Kami Ahmad, Steven Henikoff, karina_jhingan 2024. CUT&Tag Data Processing and Analysis Tutorial for Time Series. **protocols.io** https://protocols.io/view/cut-amp-tag-data-processing-and-analysis-tutorial-c86qzzdw

MANUSCRIPT CITATION: Henikoff S, Henikoff JG, Kaya-Okur HS, Ahmad K, Efficient chromatin accessibility mapping in situ by nucleosome-tethered tagmentation. eLife doi: 10.7554/eLife.63274

CUT&Tag Data Processing and Analysis Tutorial for Time Series V.1

Forked from CUT&Tag Data Processing and Analysis Tutorial

Ye Zheng[1], Kami Ahmad[1], Steven Henikoff[1], karina_jhingan[1]

[1]Fred Hutchinson Cancer Research Center

Henikovian CUT&RUNners

karina_jhingan

DISCLAIMER

Some of this writing is not mine (belonging to Ye Zheng) , and most of the code is either hers or based off of her original code.

ABSTRACT

This tutorial is largely based off of Ye Zheng's CUT&Tag analysis https://www.protocols.io/view/cut-amp-tag-data-processing-and-analysis-tutorial-e6nvw93x7gmk/v1?step=9 and has been edited to fit with time series data as well as a higher focus on producing clusters heatmaps.

Protocol status: In development
I am still doing a run through and adding in visuals

Created: Feb 13, 2024

Last Modified: Feb 27, 2024

PROTOCOL integer ID: 95152

Keywords: CUT&Tag, Data Processing, Analysis, Quality Control

I. Introduction

1 Overview of CUT&Tag

This tutorial is a time series tweaked approach based off of a CUT&Tag processing and analysis pipeline created by Ye Zheng, https://www.protocols.io/view/cut-amp-tag-data-processing-and-analysis-tutorial-e6nvw93x7gmk/v1 (**Zheng Y et al (2020). Protocol.io)**. Some sections and text from Zheng's tutorial are left unchanges. Unlike Zheng's tutorial, this pipeline will not work with duplicates and instead with time series data, focusing on the following histones: K27ac, K4me1, and K4me3.

This tutorial will not go into the science, except for the sections left in that were written by Ye Zheng, instead I'll focus on the computation/coding.

**Note: it's common to receive errors in linux if you are copying and pasting directly from this tutorial because of the format. If this happens, copy the code into a text editor and make sure the format is correct and then paste into the command line from the text editor.

2 Requirements

- Linux system
- R (versions >= 3.6)
  - dplyr

- stringr
- ggplot2
- viridis
- GenomicRanges
- chromVAR
- DESeq2
- ggpubr
- corrplot

```
## ==== R codes ==== ##
library(dplyr)
library(stringr)
library(ggplot2)
library(viridis)
library(GenomicRanges)
library(chromVAR) ## For FRiP analysis and differential analysis
library(DESeq2) ## For differential analysis section
library(ggpubr) ## For customizing figures
library(corrplot) ## For correlation plot
library(gt)
library(webshot2)
library(gtExtras)
if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
#BiocManager::install("Rsamtools")
#BiocManager::install("GenomicRanges")
#BiocManager::install("GenomicAlignments")
library(Rsamtools)
library(GenomicAlignments)
# Install this package from GitHub
#install.packages("devtools")
library(devtools)
```

- FastQC(version >= 0.11.9) [Optional]
- Bowtie2 (version >= 2.3.4.3)
- samtools (version >= 1.10)
- bedtools (version >= 2.29.1)
- Picard (version >= 2.18.29)
- SEACR (version >= 1.3)
- deepTools (version >= 2.0)

3 Data Downloading and Folder Setup

First, we need to specify the project path.

```
##== linux command ==##
mkdir /fh/fast/greenberg_p/user/kjhingan/CUT_TAG_Exp
projPath="/fh/fast/greenberg_p/user/kjhingan/CUT_TAG_Exp"
```

We have human and mice data as well as 6 different time points that data was collected from. Our analysis will be focusing on the three noted histones.

```
##== linux command ==##
histList=("K27ac" "K4me1" "K4me3")
Time=("T1" "T2" "T3" "T4" "T5" "T6")
Organism=("human" "mouse")
```

Now let's set up the folders where we will be storing the data.

```
##== linux command ==##
for org in ${Organism[@]}; do
for time in ${Time[@]}; do
for hist in ${histList[@]}; do
   mkdir -p $projPath/data/${org}/${time}/${hist}
   mkdir -p $projPath/fastq/${org}/${time}/${hist}
   mkdir -p ${projPath}/fastqFileQC/${org}/${time}/${hist}
done; done; done
```

data paths

```
human="/fh/fast/greenberg_p/SR/ngs/illumina/jwlee/220719_VH00699_162_AAC3GCV
M5/Unaligned/Project_jwlee"
mouse="/fh/fast/greenberg_p/SR/ngs/illumina/jwlee/220425_VH00699_108_AAAYMM7
M5/Unaligned/Project_jwlee"
```

And copy the data into the folders we just made (making a copy is a good idea in case we accidently alter the data files it won't alter the original data, this also allows us to organize the data into folders which makes ti easier to reference in for-loops).

```
#Human:
#T1:
cp $human/hC-T_1-H3K27ac_S3_R1_001.fastq.gz
${projPath}/data/human/T1/K27ac/R1.fastq.gz
cp $human/hC-T_1-H3K27ac_S3_R2_001.fastq.gz
${projPath}/data/human/T1/K27ac/R2.fastq.gz

cp $human/hC-T_1-H3K4me1_S4_R1_001.fastq.gz
${projPath}/data/human/T1/K4me1/R1.fastq.gz
cp $human/hC-T_1-H3K4me1_S4_R2_001.fastq.gz
${projPath}/data/human/T1/K4me1/R2.fastq.gz

cp $human/hC-T_1-H3K4me3_S6_R1_001.fastq.gz
${projPath}/data/human/T1/K4me3/R1.fastq.gz
cp $human/hC-T_1-H3K4me3_S6_R2_001.fastq.gz
${projPath}/data/human/T1/K4me3/R2.fastq.gz

#T2:
cp $human/hC-T_2-H3K27ac_S9_R1_001.fastq.gz
${projPath}/data/human/T2/K27ac/R1.fastq.gz
cp $human/hC-T_2-H3K27ac_S9_R2_001.fastq.gz
${projPath}/data/human/T2/K27ac/R2.fastq.gz

cp $human/hC-T_2-H3K4me1_S10_R1_001.fastq.gz
${projPath}/data/human/T2/K4me1/R1.fastq.gz
cp $human/hC-T_2-H3K4me1_S10_R2_001.fastq.gz
${projPath}/data/human/T2/K4me1/R2.fastq.gz

cp $human/hC-T_2-H3K4me3_S12_R1_001.fastq.gz
${projPath}/data/human/T2/K4me3/R1.fastq.gz
cp $human/hC-T_2-H3K4me3_S12_R2_001.fastq.gz
${projPath}/data/human/T2/K4me3/R2.fastq.gz

#T3:
cp $human/hC-T_3-H3K27ac_S15_R1_001.fastq.gz
${projPath}/data/human/T3/K27ac/R1.fastq.gz
cp $human/hC-T_3-H3K27ac_S15_R2_001.fastq.gz
${projPath}/data/human/T3/K27ac/R2.fastq.gz

cp $human/hC-T_3-H3K4me1_S16_R1_001.fastq.gz
${projPath}/data/human/T3/K4me1/R1.fastq.gz
cp $human/hC-T_3-H3K4me1_S16_R2_001.fastq.gz
${projPath}/data/human/T3/K4me1/R2.fastq.gz

cp $human/hC-T_3-H3K4me3_S18_R1_001.fastq.gz
${projPath}/data/human/T3/K4me3/R1.fastq.gz
```

```
cp $human/hC-T_3-H3K4me3_S18_R2_001.fastq.gz
${projPath}/data/human/T3/K4me3/R2.fastq.gz

#T4:
cp $human/hC-T_4-H3K27ac_S21_R1_001.fastq.gz
${projPath}/data/human/T4/K27ac/R1.fastq.gz
cp $human/hC-T_4-H3K27ac_S21_R2_001.fastq.gz
${projPath}/data/human/T4/K27ac/R2.fastq.gz

cp $human/hC-T_4-H3K4me1_S22_R1_001.fastq.gz
${projPath}/data/human/T4/K4me1/R1.fastq.gz
cp $human/hC-T_4-H3K4me1_S22_R2_001.fastq.gz
${projPath}/data/human/T4/K4me1/R2.fastq.gz

cp $human/hC-T_4-H3K4me3_S24_R1_001.fastq.gz
${projPath}/data/human/T4/K4me3/R1.fastq.gz
cp $human/hC-T_4-H3K4me3_S24_R2_001.fastq.gz
${projPath}/data/human/T4/K4me3/R2.fastq.gz

#T5:
cp $human/hC-T_5-H3K27ac_S27_R1_001.fastq.gz
${projPath}/data/human/T5/K27ac/R1.fastq.gz
cp $human/hC-T_5-H3K27ac_S27_R2_001.fastq.gz
${projPath}/data/human/T5/K27ac/R2.fastq.gz

cp $human/hC-T_5-H3K4me1_S28_R1_001.fastq.gz
${projPath}/data/human/T5/K4me1/R1.fastq.gz
cp $human/hC-T_5-H3K4me1_S28_R2_001.fastq.gz
${projPath}/data/human/T5/K4me1/R2.fastq.gz

cp $human/hC-T_5-H3K4me3_S30_R1_001.fastq.gz
${projPath}/data/human/T5/K4me3/R1.fastq.gz
cp $human/hC-T_5-H3K4me3_S30_R2_001.fastq.gz
${projPath}/data/human/T5/K4me3/R2.fastq.gz

#T6:
cp $human/hC-T_6-H3K27ac_S33_R1_001.fastq.gz
${projPath}/data/human/T6/K27ac/R1.fastq.gz
cp $human/hC-T_6-H3K27ac_S33_R2_001.fastq.gz
${projPath}/data/human/T6/K27ac/R2.fastq.gz

cp $human/hC-T_6-H3K4me1_S34_R1_001.fastq.gz
${projPath}/data/human/T6/K4me1/R1.fastq.gz
cp $human/hC-T_6-H3K4me1_S34_R2_001.fastq.gz
${projPath}/data/human/T6/K4me1/R2.fastq.gz
```

```
cp $human/hC-T_6-H3K4me3_S36_R1_001.fastq.gz
${projPath}/data/human/T6/K4me3/R1.fastq.gz
cp $human/hC-T_6-H3K4me3_S36_R2_001.fastq.gz
${projPath}/data/human/T6/K4me3/R2.fastq.gz



#MOUSE
#T1:
cp $mouse/mC-T_1-H3K27ac_S3_R1_001.fastq.gz
${projPath}/data/mouse/T1/K27ac/R1.fastq.gz
cp $mouse/mC-T_1-H3K27ac_S3_R2_001.fastq.gz
${projPath}/data/mouse/T1/K27ac/R2.fastq.gz

cp $mouse/mC-T_1-H3K4me1_S4_R1_001.fastq.gz
${projPath}/data/mouse/T1/K4me1/R1.fastq.gz
cp $mouse/mC-T_1-H3K4me1_S4_R2_001.fastq.gz
${projPath}/data/mouse/T1/K4me1/R2.fastq.gz

cp $mouse/mC-T_1-H3K4me3_S6_R1_001.fastq.gz
${projPath}/data/mouse/T1/K4me3/R1.fastq.gz
cp $mouse/mC-T_1-H3K4me3_S6_R2_001.fastq.gz
${projPath}/data/mouse/T1/K4me3/R2.fastq.gz

#T2:
cp $mouse/mC-T_2-H3K27ac_S9_R1_001.fastq.gz
${projPath}/data/mouse/T2/K27ac/R1.fastq.gz
cp $mouse/mC-T_2-H3K27ac_S9_R2_001.fastq.gz
${projPath}/data/mouse/T2/K27ac/R2.fastq.gz

cp $mouse/mC-T_2-H3K4me1_S10_R1_001.fastq.gz
${projPath}/data/mouse/T2/K4me1/R1.fastq.gz
cp $mouse/mC-T_2-H3K4me1_S10_R2_001.fastq.gz
${projPath}/data/mouse/T2/K4me1/R2.fastq.gz

cp $mouse/mC-T_2-H3K4me3_S12_R1_001.fastq.gz
${projPath}/data/mouse/T2/K4me3/R1.fastq.gz
cp $mouse/mC-T_2-H3K4me3_S12_R2_001.fastq.gz
${projPath}/data/mouse/T2/K4me3/R2.fastq.gz


#T3:
cp $mouse/mC-T_3-H3K27ac_S15_R1_001.fastq.gz
${projPath}/data/mouse/T3/K27ac/R1.fastq.gz
cp $mouse/mC-T_3-H3K27ac_S15_R2_001.fastq.gz
${projPath}/data/mouse/T3/K27ac/R2.fastq.gz
```

```
cp $mouse/mC-T_3-H3K4me1_S16_R1_001.fastq.gz
${projPath}/data/mouse/T3/K4me1/R1.fastq.gz
cp $mouse/mC-T_3-H3K4me1_S16_R2_001.fastq.gz
${projPath}/data/mouse/T3/K4me1/R2.fastq.gz

cp $mouse/mC-T_3-H3K4me3_S18_R1_001.fastq.gz
${projPath}/data/mouse/T3/K4me3/R1.fastq.gz
cp $mouse/mC-T_3-H3K4me3_S18_R2_001.fastq.gz
${projPath}/data/mouse/T3/K4me3/R2.fastq.gz

#T4:
cp $mouse/mC-T_4-H3K27ac_S21_R1_001.fastq.gz
${projPath}/data/mouse/T4/K27ac/R1.fastq.gz
cp $mouse/mC-T_4-H3K27ac_S21_R2_001.fastq.gz
${projPath}/data/mouse/T4/K27ac/R2.fastq.gz

cp $mouse/mC-T_4-H3K4me1_S22_R1_001.fastq.gz
${projPath}/data/mouse/T4/K4me1/R1.fastq.gz
cp $mouse/mC-T_4-H3K4me1_S22_R2_001.fastq.gz
${projPath}/data/mouse/T4/K4me1/R2.fastq.gz

cp $mouse/mC-T_4-H3K4me3_S24_R1_001.fastq.gz
${projPath}/data/mouse/T4/K4me3/R1.fastq.gz
cp $mouse/mC-T_4-H3K4me3_S24_R2_001.fastq.gz
${projPath}/data/mouse/T4/K4me3/R2.fastq.gz

#T5:
cp $mouse/mC-T_5-H3K27ac_S27_R1_001.fastq.gz
${projPath}/data/mouse/T5/K27ac/R1.fastq.gz
cp $mouse/mC-T_5-H3K27ac_S27_R2_001.fastq.gz
${projPath}/data/mouse/T5/K27ac/R2.fastq.gz

cp $mouse/mC-T_5-H3K4me1_S28_R1_001.fastq.gz
${projPath}/data/mouse/T5/K4me1/R1.fastq.gz
cp $mouse/mC-T_5-H3K4me1_S28_R2_001.fastq.gz
${projPath}/data/mouse/T5/K4me1/R2.fastq.gz

cp $mouse/mC-T_5-H3K4me3_S30_R1_001.fastq.gz
${projPath}/data/mouse/T5/K4me3/R1.fastq.gz
cp $mouse/mC-T_5-H3K4me3_S30_R2_001.fastq.gz
${projPath}/data/mouse/T5/K4me3/R2.fastq.gz

#T6:
cp $mouse/mC-T_6-H3K27ac_S33_R1_001.fastq.gz
${projPath}/data/mouse/T6/K27ac/R1.fastq.gz
cp $mouse/mC-T_6-H3K27ac_S33_R2_001.fastq.gz
${projPath}/data/mouse/T6/K27ac/R2.fastq.gz
```

```
cp $mouse/mC-T_6-H3K4me1_S34_R1_001.fastq.gz
${projPath}/data/mouse/T6/K4me1/R1.fastq.gz
cp $mouse/mC-T_6-H3K4me1_S34_R2_001.fastq.gz
${projPath}/data/mouse/T6/K4me1/R2.fastq.gz

cp $mouse/mC-T_6-H3K4me3_S36_R1_001.fastq.gz
${projPath}/data/mouse/T6/K4me3/R1.fastq.gz
cp $mouse/mC-T_6-H3K4me3_S36_R2_001.fastq.gz
${projPath}/data/mouse/T6/K4me3/R2.fastq.gz
```

II. Data Pre-processing

4 Quality Control using FastQC

1. Run FastQC for quality check

```
##== linux command ==###
ml FastQC

for org in ${Organism[@]}; do
for time in ${Time[@]}; do
for hist in ${histList[@]}; do
fastqc -o ${projPath}/fastqFileQC/${org}/${time}/${hist} -f fastq
${projPath}/data/${org}/${time}/${hist}/R1.fastq.gz
fastqc -o ${projPath}/fastqFileQC/${org}/${time}/${hist} -f fastq
${projPath}/data/${org}/${time}/${hist}/R2.fastq.gz
done; done; done
```

2. Copy over results to your local computer view the html files (alternatively open up R studio launcher for Fred Hutch and navigate to files and open through there to avoid downloading the files).

```
###== linux command (run on local shell) ==###
for org in ${Organism[@]}; do
for time in ${Time[@]}; do
for hist in ${histList[@]}; do
scp
username@rhino.fhcrc.org:$projPath/fastqFileQC/${org}/${time}/${hist}/R1_fas
tqc.html /local/path
scp
username@rhino.fhcrc.org:$projPath/fastqFileQC/${org}/${time}/${hist}/R2_fas
tqc.html /local/path
done; done; done
```

3 Interpret the quality check results.

Quality check reference:
https://www.bioinformatics.babraham.ac.uk/projects/fastqc/bad_sequence_fastqc.html.

The discordant sequence content at the beginning of the reads is a common phenomenon for CUT&Tag reads. Failing to pass the Per base sequence content does not mean your data failed.

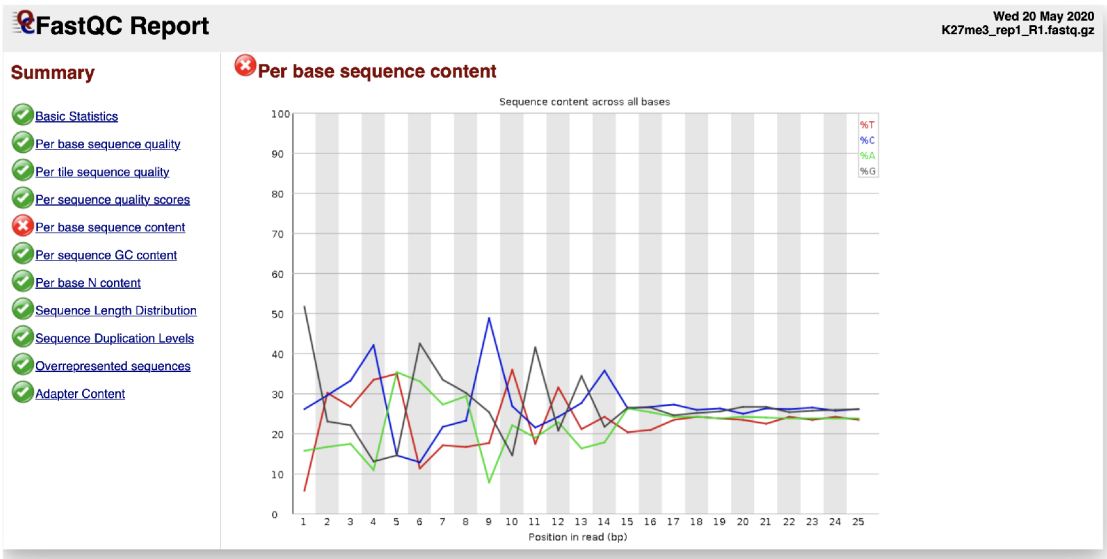


Figure 3. Per base sequence content fails the FastQC quality check.

- It can be due to the Tn5 preference.

5 Trim Reads

Documentation: https://cutadapt.readthedocs.io/en/v4.4/guide.html#paired-end
Since the experiment used 50x50 PE we will be trimming the reads with Cutadapt

```
##== linux command ==##
ml cutadapt

for org in ${Organism[@]}; do
for time in ${Time[@]}; do
for hist in ${histList[@]}; do
mkdir -p ${projPath}/data/trim/${org}/${time}/${hist}
cutadapt -o ${projPath}/data/trim/${org}/${time}/${hist}/R1.fastq -p
${projPath}/data/trim/${org}/${time}/${hist}/R2.fastq
${projPath}/data/${org}/${time}/${hist}/R1.fastq.gz
${projPath}/data/${org}/${time}/${hist}/R2.fastq.gz
done; done; done
```

example output

```
This is cutadapt 4.1 with Python 3.9.6
Command line parameters: -o
/fh/fast/greenberg_p/user/kjhingan/CUT_TAG_test/data/trim/human/T1/K27ac/R1.
fastq -p
/fh/fast/greenberg_p/user/kjhingan/CUT_TAG_test/data/trim/human/T1/K27ac/R2.
fastq
/fh/fast/greenberg_p/user/kjhingan/CUT_TAG_test/data/human/T1/K27ac/R1.fastq
.gz
/fh/fast/greenberg_p/user/kjhingan/CUT_TAG_test/data/human/T1/K27ac/R2.fastq
.gz
Processing paired-end reads on 1 core ...
Done           00:00:26     2,573,565 reads @  10.5 µs/read;   5.73 M
reads/minute
Finished in 27.08 s (11 µs/read; 5.70 M reads/minute).


=== Summary ===


Total read pairs processed:          2,573,565
Pairs written (passing filters):     2,573,565 (100.0%)


Total basepairs processed:   257,356,500 bp
  Read 1:   128,678,250 bp
  Read 2:   128,678,250 bp
Total written (filtered):    257,356,500 bp (100.0%)
  Read 1:   128,678,250 bp
  Read 2:   128,678,250 bp
```

III. Alignment

6 Bowtie2 alignment

Make folders and download genome files

```
##== linux command ==##
ml SAMtools
ml Bowtie2
mkdir $projPath/bowtie2Index
mkdir ${projPath}/bowtie2Index/hg38
mkdir ${projPath}/bowtie2Index/mm10

cd ${projPath}/data/human
wget --no-check-certificate
https://hgdownload.soe.ucsc.edu/goldenPath/hg38/bigZips/hg38.fa.gz

cd ${projPath}/data/mouse
wget --no-check-certificate
https://hgdownload.soe.ucsc.edu/goldenPath/mm10/bigZips/mm10.fa.gz
```

Build the Bowtie Index for each genome

```
##== linux command ==##
bowtie2-build ${projPath}/data/human/hg38.fa.gz
${projPath}/bowtie2Index/hg38
bowtie2-build ${projPath}/data/mouse/mm10.fa.gz
${projPath}/bowtie2Index/mm10
```

The paired-end reads are aligned by Bowtie2
*I added in the time function prior to the bowtie2 call to keep track of how long each call takes and how many calls have completed as bowtie2 does not output anything to the terminals it runs or completes.

```
##== linux command ==##
for org in ${Organism[@]}; do
if [ "${org}" == "human" ]
then
  ref="${projPath}/bowtie2Index/hg38"
else
  ref="${projPath}/bowtie2Index/mm10"
fi
for time in ${Time[@]}; do
for hist in ${histList[@]}; do
mkdir -p ${projPath}/alignment/sam/bowtie2_summary/${org}/${time}/${hist}
time bowtie2 --local --very-sensitive --no-mixed --no-discordant --phred33 -
I 10 -X 700 -p 1 -x ${ref} -1
${projPath}/data/trim/${org}/${time}/${hist}/R1.fastq -2
${projPath}/data/trim/${org}/${time}/${hist}/R2.fastq -S
${projPath}/alignment/sam/bowtie2_summary/${org}/${time}/${hist}/bowtie2.sam
&>
${projPath}/alignment/sam/bowtie2_summary/${org}/${time}/${hist}/bowtie2.txt
done; done; done
```

Note: The runtimes I was receiving were around 10-30 (a few for the mice calls took upwards of around 140 minutes) minutes for each. There are 36 calls to bowtie2 so this will take a while.

If you don't trim your reads , use this parameter instead of --local

```
--end-to-end
```

7 Alignment summary

For more detailed parameters explanation, users can refer to the bowtie2 manual

Bowtie2 alignment results summaries are saved at

```
##== linux command ==##
cat
${projPath}/alignment/sam/bowtie2_summary/${org}/${time}/${hist}/bowtie2.txt
## example below is from calling
 cat ${projPath}/alignment/sam/bowtie2_summary/human/T1/K27ac/bowtie2.txt
```

and you should expect the results look similar.

```
2573565 reads; of these:
  2573565 (100.00%) were paired; of these:
    72097 (2.80%) aligned concordantly 0 times
    1852091 (71.97%) aligned concordantly exactly 1 time
    649377 (25.23%) aligned concordantly >1 times
97.20% overall alignment rate
```

- 2573565 is the sequencing depth, i.e., the total number of paired reads.
- 72097 is the number of read-pairs that fail to be mapped.
- 1852091 + 649377 is the number of read-pairs that are successfully mapped.
- 97.20% is the overall alignment rate

III. Alignment Summary

8 Report sequencing mapping summary

Summarize the raw reads and uniquely mapping reads to report the efficiency of alignment. Alignment frequencies are expected to be >80% for high-quality data. CUT&Tag data typically has very low backgrounds, so as few as 1 million mapped fragments can give robust profiles for a histone modification in the human genome. Profiling of less-abundant transcription factors and chromatin proteins may require 10 times as many mapped fragments for downstream analysis.

We can evaluate the following metrics:

- Sequencing depth
- Alignment rate
- Number of mappable fragments
- Duplication rate
- Unique library size
- Fragment size distribution

8.1 1. Sequencing depth

```
##=== R command ===##
## Path to the project and histone list
projPath = "/fh/fast/greenberg_p/user/kjhingan/CUT_TAG_Exp"
histList = c("K27ac", "K4me1", "K4me3")
timeList = c("T1","T2","T3","T4","T5","T6")
sampleList= c("K27ac_T1", "K27ac_T2", "K27ac_T3", "K27ac_T4",
"K27ac_T5", "K27ac_T6",
              "K4me1_T1", "K4me1_T2", "K4me1_T3", "K4me1_T4",
"K4me1_T5", "K4me1_T6",
              "K4me3_T1", "K4me3_T2", "K4me3_T3", "K4me3_T4",
"K4me3_T5", "K4me3_T6")
## edit the following based on control used: we used T1 as the
control
timeControl="T1"

# this is the list of all time points except T1 that we will use
when comparing timepoints to T1
timeL = c("T2","T3","T4","T5","T6")

# this is the list of all samples except T1 that we will use when
comparing timepoints to T1
sampleControlList= c("K27ac_T2", "K27ac_T3", "K27ac_T4",
"K27ac_T5", "K27ac_T6",
                     "K4me1_T2", "K4me1_T3", "K4me1_T4",
"K4me1_T5", "K4me1_T6",
                     "K4me3_T2", "K4me3_T3", "K4me3_T4",
"K4me3_T5", "K4me3_T6")
```

I found it was simpler to just set the organism as a variable, run the entire R file (the entire R file can be found below) once with the this variable declared up at the top of the file, first organism set to mouse and then to human

You will also need to perform find and replace on all mm10/hg38 in the R file to make sure we're referencing the correct genome for the organism we're currently running the script on

```
##=== R command ===##
${projPath}/CUT_TAG_notes.R
#first run with
org="mouse"
#and then run entire program with
org="human"
```

Collect the alignment results from the bowtie2 alignment summary files

```
##=== R command ===##
alignResult = c()
for(hist in histList){
  for (time in timeList){
    alignRes = read.table(paste0(projPath,
"/alignment/sam/bowtie2_summary/", org,"/",time,"/",hist,"/",
"bowtie2.txt"), header = FALSE, fill = TRUE)
    alignRate = substr(alignRes$V1[6], 1,
nchar(as.character(alignRes$V1[6]))-1)
    alignResult = data.frame(Histone = hist, TimePoint = time,
                             SequencingDepth = alignRes$V1[1] %>%
as.character %>% as.numeric,
                             MappedFragNum_mm10 = alignRes$V1[4]
%>% as.character
                             %>% as.numeric + alignRes$V1[5] %>%
as.character %>% as.numeric,
                             AlignmentRate_mm10 = alignRate %>%
as.numeric)  %>% rbind(alignResult, .)
  }
}
alignResult$Histone = factor(alignResult$Histone, levels =
histList)
alignResult %>% mutate(AlignmentRate_mm10 =
paste0(AlignmentRate_mm10, "%"))
```

	A	B	C	D	E
	K27ac	T1	1862828	1692798	90.87%
	K27ac	T2	6024227	5830620	96.79%
	K27ac	T3	6765622	6510809	96.23%
	K27ac	T4	143632	136962	95.36%
	K27ac	T5	808940	787214	97.31%
	K27ac	T6	377282	368188	97.59%
	K4me1	T1	6073644	5928383	97.61%
	K4me1	T2	12248049	11882867	97.02%
	K4me1	T3	4995126	4800522	96.1%
	K4me1	T4	2177	2109	96.88%
	K4me1	T5	508877	500117	98.28%
	K4me1	T6	342994	336391	98.07%

	A	B	C	D	E
	K4me3	T1	13439496	13181855	98.08%
	K4me3	T2	6002937	5906183	98.39%
	K4me3	T3	4440669	4269693	96.15%
	K4me3	T4	978001	950906	97.23%
	K4me3	T5	2421522	2376602	98.14%
	K4me3	T6	2608987	2568003	98.43%

Results are from the mm10 data

2. Visualizing the sequencing depth and alignment results.

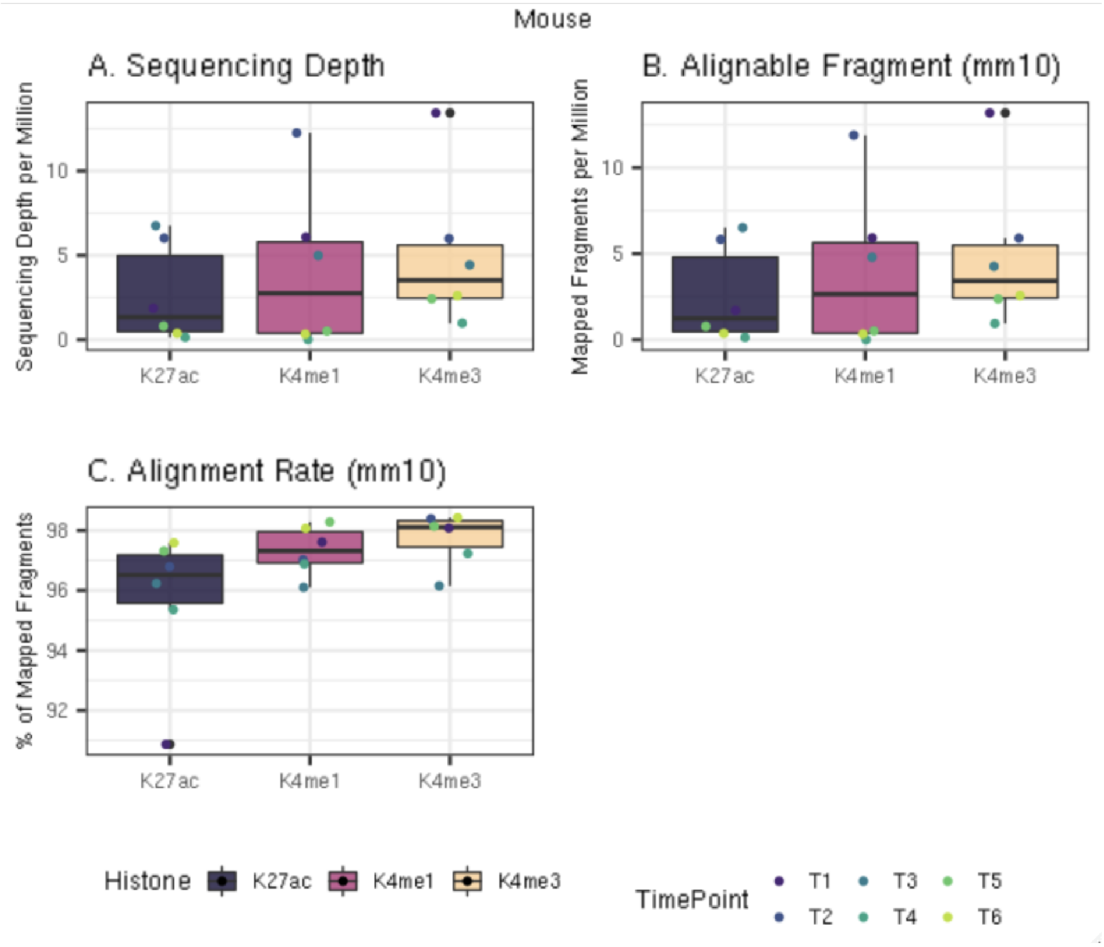
Generate sequencing depth boxplot

```
##=== R command ===##
fig3A = alignResult %>% ggplot(aes(x = Histone, y =
SequencingDepth/1000000, fill = Histone)) +
  geom_boxplot() +
  geom_jitter(aes(color = TimePoint), position =
position_jitter(0.15)) +
  scale_fill_viridis(discrete = TRUE, begin = 0.1, end = 0.9,
option = "magma", alpha = 0.8) +
  scale_color_viridis(discrete = TRUE, begin = 0.1, end = 0.9) +
  theme_bw(base_size = 18) +
  ylab("Sequencing Depth per Million") +
  xlab("") +
  ggtitle("A. Sequencing Depth") +
  theme(text = element_text(size = 12)) +
  theme(axis.title.y = element_text(size = 10))

fig3B = alignResult %>% ggplot(aes(x = Histone, y =
MappedFragNum_mm10/1000000, fill = Histone)) +
  geom_boxplot() +
  geom_jitter(aes(color = TimePoint), position =
position_jitter(0.15)) +
  scale_fill_viridis(discrete = TRUE, begin = 0.1, end = 0.9,
option = "magma", alpha = 0.8) +
  scale_color_viridis(discrete = TRUE, begin = 0.1, end = 0.9) +
  theme_bw(base_size = 18) +
  ylab("Mapped Fragments per Million") +
  xlab("") +
  ggtitle("B. Alignable Fragment (mm10)") +
  theme(text = element_text(size = 12)) +
  theme(axis.title.y = element_text(size = 10))

fig3C = alignResult %>% ggplot(aes(x = Histone, y =
AlignmentRate_mm10, fill = Histone)) +
  geom_boxplot() +
  geom_jitter(aes(color = TimePoint), position =
position_jitter(0.15)) +
  scale_fill_viridis(discrete = TRUE, begin = 0.1, end = 0.9,
option = "magma", alpha = 0.8) +
  scale_color_viridis(discrete = TRUE, begin = 0.1, end = 0.9) +
  theme_bw(base_size = 18) +
  ylab("% of Mapped Fragments") +
  xlab("") +
  ggtitle("C. Alignment Rate (mm10)") +
  theme(text = element_text(size = 12)) +
  theme(axis.title.y = element_text(size = 10))
```

```
alignPlot <- ggarrange(fig3A, fig3B, fig3C, ncol = 2, nrow=2,
common.legend = TRUE,
                       legend="bottom") %>% annotate_figure(top =
text_grob(str_to_title(org))) + bgcolor("white")
alignPlot
ggsave(plot=alignPlot,paste0(org,"_AlignmentSummaryPlot.png"),path=
paste0(projPath,"/visuals/",org))
```



IV. Alignment filtering and file format conversion

9 Prepare sam files for peak calling

Filter and keep the mapped read pairs

```
##== linux command ==##
ml SAMtools
for org in ${Organism[@]}; do
for time in ${Time[@]}; do
for hist in ${histList[@]}; do
mkdir -p $projPath/alignment/bam/${org}
samtools view -bS -F 0x04
${projPath}/alignment/sam/bowtie2_summary/${org}/${time}/${hist}/bowtie2.sam
>$projPath/alignment/bam/${org}/${time}_${hist}_bowtie2.mapped.bam
done; done; done
```

V. Peak calling

10 **MACS2**

This is assuming T1 as control as MACS2 has a control file as an input (we are performing T1 vs T2 .... T1 vs T6)

```
timeControl="T1" ;
TimeCompare=("T2" "T3" "T4" "T5" "T6");
```

parameters:
-c : control file (We are using T1 as control)
callpeak: peak calling from alignment (bowtie2) results.
-f BAMPE : our input files are formatted as BAM
-n: output file name
-q : q-value (these are the following q Values we tried (0.1, 0.001, 0.00001).
--keep-dup: we decided not to remove duplicates in this experiment (more info?)

```
##== linux command ==##
ml MACS2
for org in ${Organism[@]}; do
if [ "${org}" == "human" ]
then
   genome="hs"
else
   genome="mm"
fi
for time in ${TimeCompare[@]}; do
for hist in ${histList[@]}; do
mkdir -p
${projPath}/peakCalling/MACS2/${org}/${timeControl}_control/${time}/${hist}
macs2 callpeak -t
$projPath/alignment/bam/${org}/${time}_${hist}_bowtie2.mapped.bam \
      -c
$projPath/alignment/bam/${org}/${timeControl}_${hist}_bowtie2.mapped.bam \
      -g ${genome} -f BAMPE -n macs2_peak_q0.1 --outdir
$projPath/peakCalling/MACS2/${org}/${timeControl}_control/${time}/${hist}/
       -q 0.1 --keep-dup all
2>${projPath}/peakCalling/MACS2/${org}/${timeControl}_control/${time}/${hist
}/macs2Peak_summary.txt
done; done; done
```

add .bed to the end of your file names (as seen in the below code section) for narrow peak files in order to input into ucsc genome browser (https://genome.ucsc.edu/cgi-bin/hgCustom?hgsid=1943424434_SEeSvx8fAuJKYhm1CWSwvicOd42W) or IgV (https://igv.org/app/)

```
##== linux command ==##
for org in ${Organism[@]}; do
for time in ${TimeCompare[@]}; do
for hist in ${histList[@]}; do
mv
$projPath/peakCalling/MACS2/${org}/${timeControl}_control/${time}/${hist}/ma
cs2_peak_q0.1_peaks.narrowPeak
$projPath/peakCalling/MACS2/${org}/${timeControl}_control/${time}/${hist}/ma
cs2_peak_q0.1_peaks.narrowPeak.bed
done; done; done
```

*Note we also tested out different q values and MACS2 settings (different combinations of no model and broad) to see how it affects the outputs that we later use as the region input when creating the heatmaps

```
##== linux command ==###
## testing out different parameters for MACS2
qValues=(0.1 0.001 0.00001)
for org in ${Organism[@]}; do
if [ "${org}" == "human" ]
then
   genome="hs"
else
   genome="mm"
fi
for hist in ${histList[@]}; do
for time in ${TimeCompare[@]}; do
for q in ${qValues[@]}; do
mkdir -p outdir
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/${time}/
${hist}/

#normal
time macs2 callpeak -t
$projPath/alignment/bam/${org}/${time}_${hist}_bowtie2.mapped.bam \
      -c
$projPath/alignment/bam/${org}/${timeControl}_${hist}_bowtie2.mapped.bam \
      -g ${genome} -f BAMPE -n macs2_peak_q${q} --outdir
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/${time}/
${hist}/ \
       -q ${q} --keep-dup all

#nomodel
time macs2 callpeak -t
$projPath/alignment/bam/${org}/${time}_${hist}_bowtie2.mapped.bam \
      -c
$projPath/alignment/bam/${org}/${timeControl}_${hist}_bowtie2.mapped.bam \
      -g ${genome} -f BAMPE -n macs2_peak_q${q}_nomodel --outdir
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/${time}/
${hist}/ \
       -q ${q} --nomodel --keep-dup all

#broad
time macs2 callpeak -t
$projPath/alignment/bam/${org}/${time}_${hist}_bowtie2.mapped.bam \
      -c
$projPath/alignment/bam/${org}/${timeControl}_${hist}_bowtie2.mapped.bam \
      -g ${genome} -f BAMPE -n macs2_peak_q${q}_broad --outdir
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/${time}/
${hist}/ \
       -q ${q} --broad --keep-dup all
```

```
#nomodel and broad
time macs2 callpeak -t
$projPath/alignment/bam/${org}/${time}_${hist}_bowtie2.mapped.bam \
      -c
$projPath/alignment/bam/${org}/${timeControl}_${hist}_bowtie2.mapped.bam \
      -g ${genome} -f BAMPE -n macs2_peak_q${q}_nomodel_broad --outdir
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/${time}/
${hist}/ \
       -q ${q} --broad --nomodel --keep-dup all
done; done; done; done
```

11 FRagment proportion in Peaks regions (FRiPs)

We calculate the fraction of reads in peaks (FRiPs) as a measure of signal-to-noise and contrast it to FRiPs in the IgG control dataset for illustration. Although sequencing depths for CUT&Tag are typically only 1-5 million reads, the low background of the method results in high FRiP scores.

```
##=== R command ===##
peakN = c()
#peakWidth = c()
peakOverlap = c()
inPeakData = c()
for(sample in sampleControlList){
  histInfo=str_split(sample,"_")[[1]]
  hist=histInfo[1]
  time=histInfo[2]
  peakInfo = read.table(paste0(projPath,
"/peakCalling/MACS2/",org,"/",timeControl,"_control/",
                               time,"/",hist,"/",
"macs2_peak_q0.1_peaks.narrowPeak.bed"),
                        header = FALSE, fill = TRUE)  %>% mutate(width =
abs(V3-V2))
  peakN = data.frame(Histone = hist, TimePoint = time, peakN =
nrow(peakInfo)) %>% rbind(peakN, .)
  #peakWidth = data.frame(Histone = hist,TimePoint = time,
  #                       width = peakInfo$width)  %>% rbind(peakWidth, .)
  peak.gr = GRanges(seqnames = peakInfo$V1, IRanges(start = peakInfo$V2,
                                                    end = peakInfo$V3),
strand = "*")
  bamFile = paste0(projPath,"/alignment/bam/",
org,"/",time,"/",hist,"/bowtie2.mapped.bam")
  fragment_counts <- getCounts(bamFile, peak.gr, paired = TRUE, by_rg =
FALSE,
                               format = "bam")
  inPeakN = counts(fragment_counts)[,1] %>% sum
  inPeakData = rbind(inPeakData, data.frame(inPeakN = inPeakN, Histone =
hist,
                                            TimePoint = time))
}
head(inPeakData)
peakN %>% select(Histone, TimePoint, peakN)

frip = left_join(inPeakData, alignResult, by = c("Histone", "TimePoint"))
%>%
  mutate(frip = inPeakN/MappedFragNum_mm10 * 100)
frip %>% select(Histone, TimePoint, SequencingDepth, MappedFragNum_mm10,
                AlignmentRate_mm10, FragInPeakNum = inPeakN, FRiPs = frip)
```

12 **Visualization of peak number, peak width, and FRiPs**

```
##=== R command ===##
fig7A = peakN %>% ggplot(aes(x = Histone, y = peakN, fill = Histone)) +
  geom_boxplot() +
  geom_jitter(aes(color = TimePoint), position = position_jitter(0.15)) +
  scale_fill_viridis(discrete = TRUE, begin = 0.1, end = 0.55, option =
"magma", alpha = 0.8) +
  scale_color_viridis(discrete = TRUE, begin = 0.1, end = 0.9) +
  theme_bw(base_size = 10) +
  ylab("Number of Peaks") +
  xlab("") +
  theme(axis.text=element_text(size=8), axis.title.y=element_text(size=12))

fig7B = peakN %>% ggplot(aes(x = Histone, y = peakWidth, fill = Histone)) +
  geom_violin() +
  #facet_grid(TimePoint) +
  scale_fill_viridis(discrete = TRUE, begin = 0.1, end = 0.55, option =
"magma", alpha = 0.8) +
  scale_color_viridis(discrete = TRUE, begin = 0.1, end = 0.9) +
  scale_y_continuous(trans = "log", breaks = c(400, 3000, 22000)) +
  theme_bw(base_size = 10) +
  ylab("Width of Peaks") +
  xlab("") +
  theme(axis.text=element_text(size=8), axis.title.y=element_text(size=12))

fig7D = frip %>% ggplot(aes(x = Histone, y = frip, fill = Histone, label =
round(frip, 2))) +
  geom_boxplot() +
  geom_jitter(aes(color = TimePoint), position = position_jitter(0.15)) +
  scale_fill_viridis(discrete = TRUE, begin = 0.1, end = 0.55, option =
"magma", alpha = 0.8) +
  scale_color_viridis(discrete = TRUE, begin = 0.1, end = 0.9) +
  theme_bw(base_size = 18) +
  ylab("% of Fragments in Peaks") +
  xlab("") +
  theme(axis.text=element_text(size=8), axis.title.y=element_text(size=12))
PeakPlot <- ggarrange(fig7A,fig7D, common.legend = TRUE, legend="bottom")
%>%
  annotate_figure(top = text_grob(str_to_title(org))) + bgcolor("white")
PeakPlot
```

VI. File Formatting

13 **More FIle Format Conversion**

Here we will sort our bam files by coordinates.

```
###== linux command ==###
mkdir -p $projPath/alignment/bigwig
samtools sort -o $projPath/alignment/bam/${histName}.sorted.bam
$projPath/alignment/bam/${histName}_bowtie2.mapped.bam
samtools index $projPath/alignment/bam/${histName}.sorted.bam
```

VII. Scaling

14 The original tutorial scales the data using E.coli counts . Since our data did not contain E.coli we will be scaling based on reads per genome coverage.

Here we are going to be scaling the bigwig files that we will be inputting to create our heatmaps.

*gzsize values can be found in deeptools bamcoverage documentation
https://deeptools.readthedocs.io/en/latest/content/feature/effectiveGenomeSize.html

```
###== linux command ==###
for org in ${Organism[@]}; do
if [ "${org}" == "human" ]
then
   gsize="2913022398"
else
   gsize="2652783500"
fi
for time in ${Time[@]}; do
for hist in ${histList[@]}; do
bamCoverage --normalizeUsing RPGC --effectiveGenomeSize ${gsize} -b
$projPath/alignment/bam/${org}/${time}/${hist}/sorted.bam -o
$projPath/alignment/bigwig/${org}/${time}_${hist}_normalized.bw
done; done; done
```

VIII. Visualization

15 **Heatmap Setup**

Setup for using DeepTool's heatmap functions

```
##== linux command ==###
ml R
ml SAMtools
ml Homer
ml deepTools
ml SAMtools
```

*Note: if you receive an error that the mm10 genome is not available through the Homer program loaded through 'ml Homer'. You may need to download the Homer program

```
##== linux command (if needed)==###
#if you receive an error about a genome not found and are unable to install
it you will have to #download the package yourself if you are using a
cluster computing server where you #loaded homer via module load
#installing HOMER as a package
cd $projPath/tools
mkdir $projPath/tools/Homer
wget http://homer.ucsd.edu/homer/configureHomer.pl
perl $projPath/tools/Homer/configureHomer.pl -install
perl $projPath/tools/Homer/configureHomer.pl -install mm10
perl $projPath/tools/Homer/configureHomer.pl -install hg38
PATH=$PATH:$projPath/tools/Homer/bin/
#create txt file with "PATH=$PATH:$projPath/tools/Homer/bin/" and move into
projPath #directory and name bash_profile
##end of installation
```

and run this following code in place of 'ml Homer'

```
##== linux command (if needed)==###
cd $projPath/tools/Homer/bin
source $projPath/bash_profile
```

16 Heatmap on CUT&Tag MACS2 peaks

This is a large for-loop so for ease I will be breaking the code block into steps explaining what is going on.

An overview of the code is that in order to have the heatmap focus on the regions where we have the MACS2 peaks we will be merging the peak files using Homer's mergePeaks. mergePeak's output is a txt file,

so we will turn it into a bed file and then use that as the input for the -R (region) parameter in deepTool's heatmap.

Step 1) Merge Peaks: We tried using various values for distances (the best value will likely vary for different histones and genomes); distances=("5000" "7500" "10000"). Here we merge all of our peakCalling data for the given Histone and genome

Step 2) Convert mergePeak output into a bed file

Step 3) This is the actual heatmap function, or at least the first part. As creating the heatmap involves first creating the matrix and then the second part is plotting the matrix.
parameters:
-p 10: number of cores (this is a timely function so I would recommend instead of running the entire for loop at once, try just running pick an example genome, histone, and distance combination to run once to make sure it runs alright (the most common error is referencing a file that does not exist so check your input files for typos). It is normal for either the matrix or plotting portion to sometimes take up to 12 hours for one function call.

Step 4) This is where we input the matrix file we created form the compute matrix function and output the heat map visual, we also have the option to save the cluster information in outFileSortedRegions. We will be using the cluster information to run Homer's motif finding later to analyze each cluster. The only parameter that needs tuning here is the kmeans parameter as this tells the function how many clusters to look for. We tried clusterSizes=("3" "4" "5")

Parameter values to try out

```
##== linux command ==##
distances=("5000" "7500" "10000")
clusterSizes=("3" "4" "5")
```

```
##== linux command ==###
for org in ${Organism[@]}; do
for hist in ${histList[@]}; do
for distance in ${distances[@]}; do

##Step 1
mergePeaks -d ${distance}
$projPath/peakCalling/MACS2/${timeControl}_control/${org}_T2_${hist}_macs2_p
eak_q0.1_peaks.narrowPeak.bed
$projPath/peakCalling/MACS2/${timeControl}_control/${org}_T3_${hist}_macs2_p
eak_q0.1_peaks.narrowPeak.bed
$projPath/peakCalling/MACS2/${timeControl}_control/${org}_T4_${hist}_macs2_p
eak_q0.1_peaks.narrowPeak.bed
$projPath/peakCalling/MACS2/${timeControl}_control/${org}_T5_${hist}_macs2_p
eak_q0.1_peaks.narrowPeak.bed
$projPath/peakCalling/MACS2/${timeControl}_control/${org}_T6_${hist}_macs2_p
eak_q0.1_peaks.narrowPeak.bed
>$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_mergedPeakFile.txt

##Step 2
awk 'BEGIN { OFS="\t" } {
    print $2, $3, $4, $1;
}'
"$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_mergedPeakFile.txt" >
"$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_mergedPeakFile.bed"
echo "$(tail -n +2
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_mergedPeakFile.bed)" >
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_mergedPeakFile.bed

##Step 3
computeMatrix scale-regions -S
$projPath/alignment/bigwig/${org}/T1_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T2_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T3_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T4_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T5_${hist}_normalized.bw \
```

```
$projPath/alignment/bigwig/${org}/T6_${hist}_normalized.bw \
                                -R
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_mergedPeakFile.bed \
                                --skipZeros --missingDataAsZero  -o
$projPath/heatmap/${org}/${hist}/${distance}_cluster_matrix_gene.mat.gz -p
10
for cluster in ${clusterSizes[@]}; do

##Step 4
plotHeatmap -m
$projPath/heatmap/${org}/${hist}/${distance}_cluster_matrix_gene.mat.gz -out
$projPath/heatmap/${org}/${hist}/${distance}_${cluster}_Histone_gene.png --
kmeans ${cluster} --outFileSortedRegions
${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_info.bed

done; done; done; done
```

*To continue the different combinations we tried in the Alignment MACS2 step where we tried different combinations of no model and broad, as well as the different q values; this is the script to run the heatmaps on all the various outputs

Caution This will take a very long time to run. I recommend only tuning one parameter at a time (i.e remove one of the for x in xlist just set that value to one, for example replace 'for distance in ${distances[@]}; do' with distance = "5000" to try out the different q values)

```
##== linux command ==##
for org in ${Organism[@]}; do
for hist in ${histList[@]}; do
for distance in ${distances[@]}; do
mkdir -p $projPath/heatmap/parameters/${org}/${hist}/
for q in ${qValues[@]}; do
time mergePeaks -d ${distance}
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T2/${his
t}/macs2_peak_q${q}_peaks.narrowPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T3/${his
t}/macs2_peak_q${q}_peaks.narrowPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T4/${his
t}/macs2_peak_q${q}_peaks.narrowPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T5/${his
t}/macs2_peak_q${q}_peaks.narrowPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T6/${his
t}/macs2_peak_q${q}_peaks.narrowPeak
>$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_q${q}_mergedPeakFile.txt
awk 'BEGIN { OFS="\t" } {
    print $2, $3, $4, $1;
}'
"$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_q${q}_mergedPeakFile.txt" >
"$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_q${q}_mergedPeakFile.bed"
echo "$(tail -n +2
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_q${q}_mergedPeakFile.bed)" >
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_q${q}_mergedPeakFile.bed
time computeMatrix scale-regions -S
$projPath/alignment/bigwig/${org}/T1_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T2_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T3_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T4_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T5_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T6_${hist}_normalized.bw \
                              -R
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_q${q}_mergedPeakFile.bed \
```

```
                                --skipZeros --missingDataAsZero  -o
$projPath/heatmap/parameters/${org}/${hist}/${distance}_q${q}_cluster_matrix
_gene.mat.gz -p 10
# nomodel
time mergePeaks -d ${distance}
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T2/${his
t}/macs2_peak_q${q}_nomodel_peaks.narrowPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T3/${his
t}/macs2_peak_q${q}_nomodel_peaks.narrowPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T4/${his
t}/macs2_peak_q${q}_nomodel_peaks.narrowPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T5/${his
t}/macs2_peak_q${q}_nomodel_peaks.narrowPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T6/${his
t}/macs2_peak_q${q}_nomodel_peaks.narrowPeak
>$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_q${q}_nomodel_mergedPeakFile.txt
awk 'BEGIN { OFS="\t" } {
    print $2, $3, $4, $1;
}'
"$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_q${q}_nomodel_mergedPeakFile.txt" >
"$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_q${q}_nomodel_mergedPeakFile.bed"
echo "$(tail -n +2
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_q${q}_nomodel_mergedPeakFile.bed)" >
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_q${q}_nomodel_mergedPeakFile.bed
time computeMatrix scale-regions -S
$projPath/alignment/bigwig/${org}/T1_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T2_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T3_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T4_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T5_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T6_${hist}_normalized.bw \
                                -R
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_q${q}_nomodel_mergedPeakFile.bed \
                                --skipZeros --missingDataAsZero  -o
$projPath/heatmap/parameters/${org}/${hist}/${distance}_q${q}_nomodel_cluste
r_matrix_gene.mat.gz -p 10
```

```
# broad
time mergePeaks -d ${distance}
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T2/${his
t}/macs2_peak_q${q}_broad_peaks.broadPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T3/${his
t}/macs2_peak_q${q}_broad_peaks.broadPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T4/${his
t}/macs2_peak_q${q}_broad_peaks.broadPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T5/${his
t}/macs2_peak_q${q}_broad_peaks.broadPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T6/${his
t}/macs2_peak_q${q}_broad_peaks.broadPeak
>$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_q${q}_broad_mergedPeakFile.txt
awk 'BEGIN { OFS="\t" } {
    print $2, $3, $4, $1;
}'
"$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_q${q}_broad_mergedPeakFile.txt" >
"$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_q${q}_broad_mergedPeakFile.bed"
echo "$(tail -n +2
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_q${q}_broad_mergedPeakFile.bed)" >
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_q${q}_broad_mergedPeakFile.bed
time computeMatrix scale-regions -S
$projPath/alignment/bigwig/${org}/T1_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T2_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T3_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T4_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T5_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T6_${hist}_normalized.bw \
                                -R
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_q${q}_broad_mergedPeakFile.bed \
                                --skipZeros --missingDataAsZero  -o
$projPath/heatmap/parameters/${org}/${hist}/${distance}_q${q}_broad_cluster_
matrix_gene.mat.gz -p 10
# broad nomodel
time mergePeaks -d ${distance}
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T2/${his
```

```
t}/macs2_peak_q${q}_nomodel_broad_peaks.broadPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T3/${his
t}/macs2_peak_q${q}_nomodel_broad_peaks.broadPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T4/${his
t}/macs2_peak_q${q}_nomodel_broad_peaks.broadPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T5/${his
t}/macs2_peak_q${q}_nomodel_broad_peaks.broadPeak
$projPath/peakCalling/MACS2/${org}/parameter/${timeControl}_control/T6/${his
t}/macs2_peak_q${q}_nomodel_broad_peaks.broadPeak
>$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_q${q}_nomodel_broad_mergedPeakFile.txt
awk 'BEGIN { OFS="\t" } {
    print $2, $3, $4, $1;
}'
"$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_q${q}_nomodel_broad_mergedPeakFile.txt" >
"$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance
}_q${q}_nomodel_broad_mergedPeakFile.bed"
echo "$(tail -n +2
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_q${q}_nomodel_broad_mergedPeakFile.bed)" >
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_q${q}_nomodel_broad_mergedPeakFile.bed
time computeMatrix scale-regions -S
$projPath/alignment/bigwig/${org}/T1_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T2_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T3_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T4_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T5_${hist}_normalized.bw \

$projPath/alignment/bigwig/${org}/T6_${hist}_normalized.bw \
                              -R
$projPath/Homer/mergePeaks/${timeControl}_control/${org}_${hist}_${distance}
_q${q}_nomodel_broad_mergedPeakFile.bed \
                              --skipZeros --missingDataAsZero  -o
$projPath/heatmap/parameters/${org}/${hist}/${distance}_q${q}_nomodel_broad_
cluster_matrix_gene.mat.gz -p 10
for cluster in ${clusterSizes[@]}; do
time plotHeatmap -m
$projPath/heatmap/parameters/${org}/${hist}/${distance}_q${q}_cluster_matrix
_gene.mat.gz -out
$projPath/heatmap/parameters/${org}/${hist}/${distance}_${cluster}_q${q}_His
tone_gene.png --kmeans ${cluster} --outFileSortedRegions
```

```
${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_q${q}_info.bed
time plotHeatmap -m
$projPath/heatmap/parameters/${org}/${hist}/${distance}_q${q}_nomodel_cluste
r_matrix_gene.mat.gz -out
$projPath/heatmap/parameters/${org}/${hist}/${distance}_${cluster}_q${q}_nom
odel_Histone_gene.png --kmeans ${cluster} --outFileSortedRegions
${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_q${q}_nomodel_
info.bed
time plotHeatmap -m
$projPath/heatmap/parameters/${org}/${hist}/${distance}_q${q}_broad_cluster_
matrix_gene.mat.gz -out
$projPath/heatmap/parameters/${org}/${hist}/${distance}_${cluster}_q${q}_bro
ad_Histone_gene.png --kmeans ${cluster} --outFileSortedRegions
${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_q${q}_broad_in
fo.bed
time plotHeatmap -m
$projPath/heatmap/parameters/${org}/${hist}/${distance}_q${q}_nomodel_broad_
cluster_matrix_gene.mat.gz -out
$projPath/heatmap/parameters/${org}/${hist}/${distance}_${cluster}_q${q}_nom
odel_broad_Histone_gene.png --kmeans ${cluster} --outFileSortedRegions
${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_q${q}_nomodel_
broad_info.bed

done; done; done; done; done
```

VIII. Analyze Heatmap Cluster

17 **Split up bed file into clusters**

The bed file from plotHeatmap outputs information where there's a column with the cluster number. we want to separate the bed file based on the cluster column so that we have one bed file per cluster and we can analyze the clusters separately.

How you run this next code depends on the value chosen for kmeans in plotHeatmap, for example if kmeans is 3 you would set

```
##== linux command ==##
cluster="3"
```

and only run up until and including clusterGroup=("3") as we would only have 3 clusters to split the bed file into. (And if kmeans=4, then cluster="4" and run up to and including clusterGroup=("6"), and so on for other kmeans values)

```
##== linux command ==###
clusterGroup=("1")
for org in ${Organism[@]}; do
for distance in ${distances[@]}; do
for hist in ${histList[@]}; do
awk '$13=="cluster_1"'
${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_info.bed
>${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_${clusterGrou
p}_info.bed
done; done; done

clusterGroup=("2")
for org in ${Organism[@]}; do
for distance in ${distances[@]}; do
for hist in ${histList[@]}; do
awk '$13=="cluster_2"'
${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_info.bed
>${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_${clusterGrou
p}_info.bed
done; done; done

clusterGroup=("3")
for org in ${Organism[@]}; do
for distance in ${distances[@]}; do
for hist in ${histList[@]}; do
awk '$13=="cluster_3"'
${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_info.bed
>${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_${clusterGrou
p}_info.bed
done; done; done

## stop here if kmeans=3

cluster=("4")
for org in ${Organism[@]}; do
for distance in ${distances[@]}; do
for hist in ${histList[@]}; do
awk '$13=="cluster_4"'
${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_info.bed
>${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_${clusterGrou
p}_info.bed
done; done; done

## stop here if kmeans=4
cluster=("5")
for org in ${Organism[@]}; do
```

```
for distance in ${distances[@]}; do
for hist in ${histList[@]}; do
awk '$13=="cluster_5"'
${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_info.bed
>${projPath}/clusterInfo/${org}/${hist}/${distance}_${cluster}_${clusterGrou
p}_info.bed
done; done; done

## stop here if kmeans=5
```