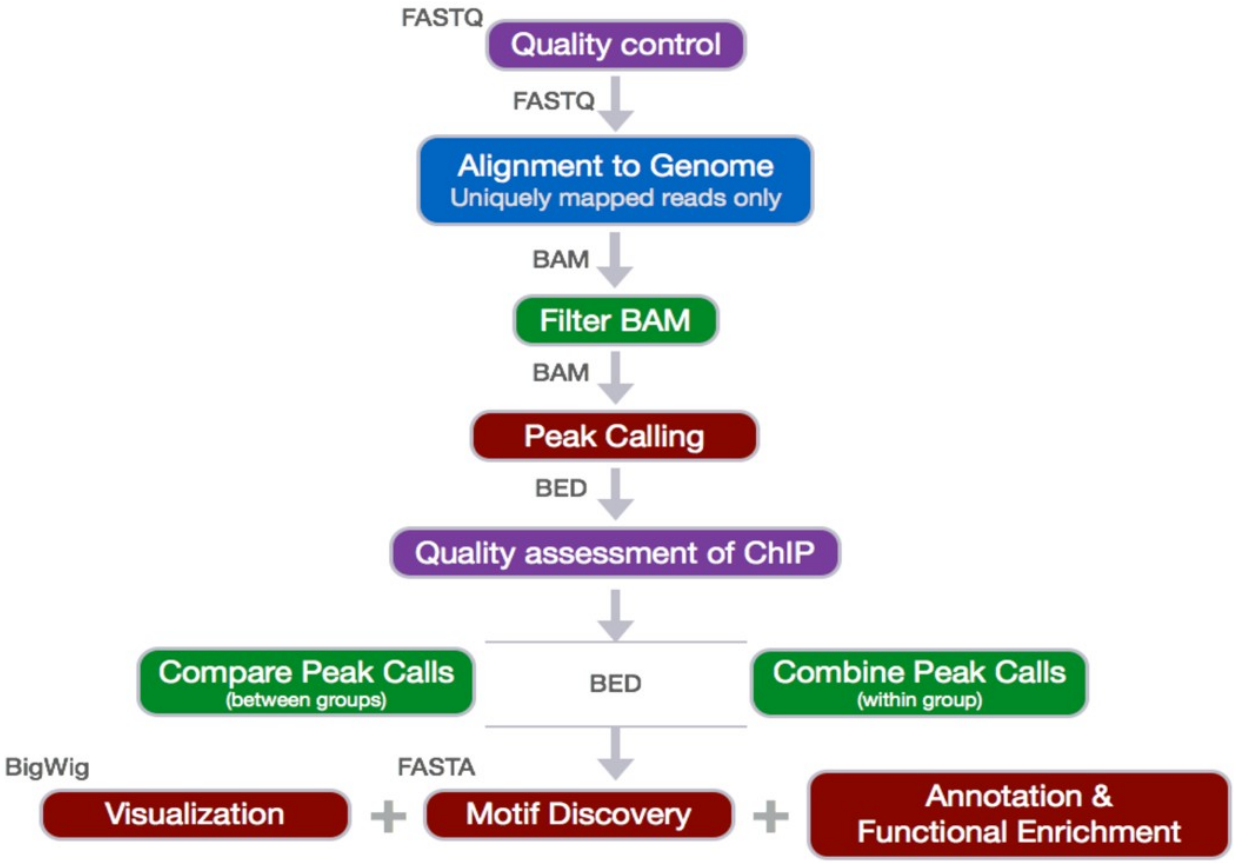
CHIP-SEQ DATA ANALYSIS WORKFLOW



- Goal:Chromatin immunoprecipitation (ChIP) experiments are performed to identify DNA that binds specific (chromatin) proteins of interest.

Data Analysis Pipeline Steps:

a. Sequence Reads

b. Quality Control (Evaluate the quality of the sequencing data)à Fastqc

c. Alignment to Genome & Result Output to SAM à Bowtie2

d.SAM to BAM conversion à Samtools

e. Sorting BAM by Genomic Coordinates and filtering only uniquely mapped reads à Sambamba

f. Peak Calling (Identify **areas** in the genome that have been enriched with **aligned reads** as a consequence of performing a ChIP-Seq experiment) à MACS

g. Downstream Analysis (annotation of the peaks)

Organism: Oryza sativa
Secondary Study Accession: SRP017137
Study Title:
Chromatin immunoprecipitation of MADS29 to find the cis regulatory regions and targets of this transcription factor

Source

- DNA(3)

Library Layout

- single(3)

Platform

- Illumina(3)

Strategy

- other(3)

TOTAL biosample accession no. =

SRR616062
SRR616063
SRR616064

Project: PRJNA179281

A ChIP-seq analysis revealed 378 targets of OsMADS29, which include genes involved in cytokinin metabolism and auxin signaling, carbohydrate metabolism, transporters and a large number of transcription factors, reflecting on its functional diversity. Overall design: Chromatin Immunonoprecipitation of MADS29 using antiMADS29 antibody was done using rice 9 DAP seed tissue with total input chromatin and mock immunoprecipitated chromatin as controls

#FastQC
sudo apt-get install fastqc

#Trimmomatic
sudo apt get update
sudo apt-get install trimmomatic-0.36

for single end=
~/Documents/chip_seq/ena_files(path)
java -jar trimmomatic-0.36.jar SE SRR616062/SRR616062.fastq.gz forward62.fq ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:30 MINLEN:20
(output= TrimmomaticSE: Started with arguments:
SRR616062/SRR616062.fastq.gz forward62.fa ILLUMINACLIP:TruSeq3-SE.fa:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:30 MINLEN:20
Automatically using 1 threads

Using Long Clipping Sequence: 'AGATCGGAAGAGCGTCGTGTAGGGAAAGAGTGTA'
Using Long Clipping Sequence: 'AGATCGGAAGAGCACACGTCTGAACTCCAGTCAC'
ILLUMINACLIP: Using 0 prefix pairs, 2 forward/reverse sequences, 0 forward only sequences, 0 reverse only sequences
Quality encoding detected as phred33
Input Reads: 11791914 Surviving: 10076007 (85.45%) Dropped: 1715907 (14.55%)
TrimmomaticSE: Completed successfully)
(create a forward62.fa)

#bowtie2

sudo apt-get install bowti2
The bowtie2-build indexer-
Command Line

Usage:

#bowtie2-build [options]* <reference_in> <bt2_base>
bowtie2-build -f GCF_001433935.1_IRGSP-1.0_genomic.fna.gz b2_base
bowtie2-build builds a Bowtie index from a set of DNA sequences. bowtie2-build outputs a set of 6 files with suffixes.
(create file =(1) b2_base.1.bt2 , (2) b2_base.2.bt2, (3) b2_base.3.bt2, (4)b2_base.4.bt2, (5)b2_base.rev.1.bt2 , (6)b2_base.rev.2.bt2

#Getting started with Bowtie 2=
bowtie2 -x b2_base -U forward62.fq -S forward62.sam(open terminal where all index file)
(output- 0076007 reads; of these:
  10076007 (100.00%) were unpaired; of these:
    2921040 (28.99%) aligned 0 times
    4355722 (43.23%) aligned exactly 1 time
    2799245 (27.78%) aligned >1 times
71.01% overall alignment rate)
head forward62.sam(cammands)
(outputs=
@HD VN:1.0 SO:unsorted
@SQ SN:NC_029256.1 LN:43270923
@SQ SN:NC_029257.1 LN:35937250
@SQ SN:NC_029258.1 LN:36413819
@SQ SN:NC_029259.1 LN:35502694
@SQ SN:NC_029260.1 LN:29958434
@SQ SN:NC_029261.1 LN:31248787
@SQ SN:NC_029262.1 LN:29697621
@SQ SN:NC_029263.1 LN:28443022
@SQ SN:NC_029264.1 LN:23012720)
 bowtie2 -x b2_base -U forward63.fq -S forward63.sam
(output=33812514 reads; of these:
  33812514 (100.00%) were unpaired; of these:
    2809217 (8.31%) aligned 0 times
    16956308 (50.15%) aligned exactly 1 time
    14046989 (41.54%) aligned >1 times
91.69% overall alignment rate)
bowtie2 -x b2_base -U forward64.fq -S forward64.sam

(output= e2 -x b2_base -U forward64.fq -S forward64.sam
137885 reads; of these:
  137885 (100.00%) were unpaired; of these:
    11074 (8.03%) aligned 0 times
    69871 (50.67%) aligned exactly 1 time
    56940 (41.30%) aligned >1 times
91.97% overall alignment rate)
head forward62.sam
head -20 forward62.sam(open sam file)
#SAM TO BAM
IN SAM file(SRR616062.10525368 0 NC_029256.1 5838198 42 36M * 00
GGAAGCTCTGGTGGAGCTGCAGCGCGTGCGCGCCGT
IIIIIIIIIIHIIHIIIIIGIIIIIHIIIIIIIIG AS:i:0 X)

#install samtools

The samtools help= (samtools view ,samtools sort ,samtools depth|)
#Converting SAM to BAM with samtools “view"
samtools view -S -b sample.sam > sample.bam
samtools view sample.bam | head
#samtools "sort"
samtools view sample.bam | head
samtools sort sample.bam -o sample.sorted.bam
#Now let’s check the order:
samtools view sample.sorted.bam | head
samtools view sample.sorted.bam | wc -l(#count total alignment= 10076007)
samtools view -H sample.sorted.bam(inspect header)
samtools view forward62.sorted.bam | awk '{print $5}' | sort -k1g| uniq -c
samtools view forward62.sorted.bam | awk '{print $2}' | sort | uniq -c
samtools view -f 0*2 forward62.sorted.bam |head(capture the FLAG)
samtools view -f 0*2 forward62.sorted.bam |wc -l(how mony properly alignment there)

#install sambamba tool(filtering bam file(remove time alignment sequence, remove >1 time
alignment sequence we should only one time sequence alignment)
sambamba-0.8.0-linux-amd64-static.gzsambamba-0.8.0-linux-amd64-static.gz
chmod u+x sambamba-0.8.0
./sambamba-0.8.0
Available commands:

```
  view       view contents and convert from one format
             to another (SAM/BAM/JSON/UNPACK)
  index      build index (BAI)
  merge      merge files (BAM)
  sort       sort file (BAM)
  slice      slice file (BAM using BED)
  markdup    mark or remove duplicates (BAM)
  subsample  subsample (BAM)
  flagstat   output statistics (BAM)
  depth      output statistics (BAM)
  validate   simple validator (BAM)
```

./sambamba-0.8.0 -h
./sambamba-0.8.0 view forward64.bam | head |wc -l(# total world 8868193)

./sambamba-0.8.0 view forward64.bam | head |wc

./sambamba-0.8.0 view forward64.bam | head
sambamba markdup OPTIONS <input.bam> <output.bam>(./sambamba-0.8.0 markdup -r forward62.bam output62.bam)
./sambamba-0.8.0 markdup -r forward63.bam output63.bam(remove dublicate aligned sequence)
./sambamba-0.8.0 markdup -r forward62.bam markdup62.bam(markdup.bam output file)
(output= inding positions of the duplicate reads in the file...
  sorted 0 end pairs
     and 31003297 single ends (among them 0 unmatched pairs)
  collecting indices of duplicate reads...   done in 3580 ms
  found 9919438 duplicates
collected list of positions in 0 min 32 sec
removing duplicates...
collected list of positions in 0 min 50 sec)
#MACS2(INSTALL MACS2 tools cammands and requirement tools)
Requirements
1.1.1 Software
• Python 2.7.10+
• Cython 0.25+
• HTSlib

• SAMtools
(sudo apt-get install make build-essential zlib1g-dev libbz2-dev liblzma-dev curl pigz
‹ →libncurses5-dev)
pip install Cython
git clone https://github.com/samtools/htslib.git
cd htslib
autoheader
autoconf

MACS2 installation
git clone https://github.com/Multiscale-Genomics/mg-process-macs2.git
pip install git+https://github.com/Multiscale-Genomics/mg-process-macs2.git
Usage: macs2 <-t tfile> [-n name] [-g genomesize] [options]

Example: macs2 -t ChIP.bam -c Control.bam -f BAM -g hs -n test -B -q 0.01
macs2 callpeak -t output62.bam -f BAM -g 4.3e+6 --nomodel -n test -B --outdir macs2_1 -q 0.01
samtools sort output62.bam -o output62.sorted.bam
macs2 callpeak -t uniq62.bam -f BAM --name=peak1q -q 0.01(file create) or macs2 callpeak -t uniq62.bam -f BAM --name=peak1q --outdir macs2_62(file name) -q 0.01
 Rscript NAME_model.r
 Rscript peak1_model.r(create model.r)
 Rscript peak1q_model.r(all output files start peak1q_)
macs2 callpeak -t uniq64.bam -f BAM --name=peak1q --outdir macs2_64 -q 0.01
Rscript peak1q_model.r
macs2 callpeak -t uniq63.bam -f BAM --name=peak1q --outdir macs2_63 -q 0.01
Rscript peak1q_model.r(open macs2_63 file then use cammand)