Changes in chromatin accessibility between Arabidopsis stem cells and mesophyll cells illuminate cell type-specific transcription factor networks

Basic Information: IF: 6.2 *the* plant journal

1 ATAC-seq 原理

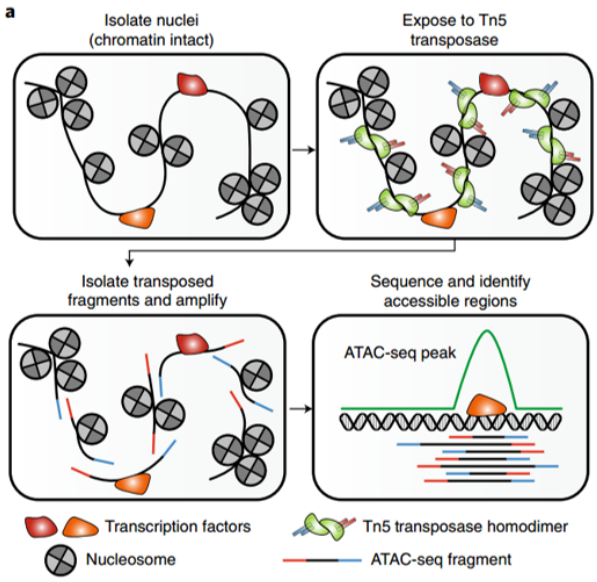


Fig. 1 ATAC-seq 实验原理

染色质可及性表示染色质开放程度的大小，染色质开放，调控因子就可以结合在对应位置并调节基因的表达，沉默。在干细胞的分化过程中，基因往往处于一种就绪的状态。在该过程中基因的开放程度高，此时一些调控因子可以结合在相应的位点，调控基因的开或者关。基因的开或者关往往决定着细胞的分化方向。基因组结构和染色质动态性控制着细胞命运 [1]。

ATAC-seq 使用 Tn5 处理，Tn5 可以结合在染色质开放的位置，并且切割该位置，添加一段已知的序列 [2]。测序并且 align 后的结果会在两侧出现峰的信号。(信号都是 paired 的，非配对的，只有一侧的会被忽略掉。) 但是

align 出来的峰值不是真正的 Tn5 切割的位置，真正切割的位置在序列的开头，因此要进行移峰的操作，使得峰值真正地代表染色质开放位置。

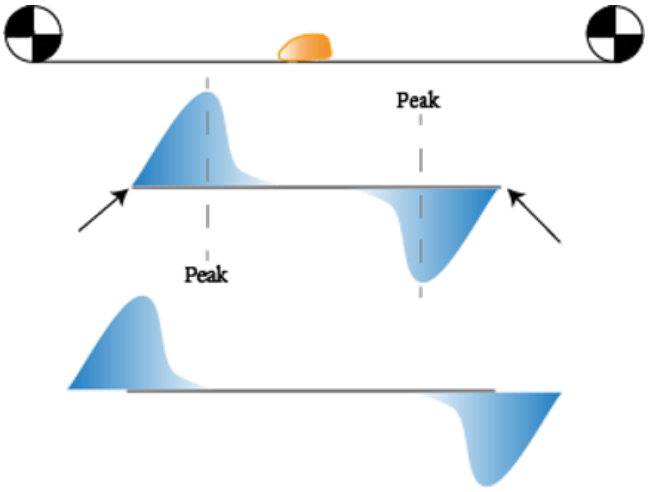


Fig. 2 ATAC-seq Peakcall 原理

下图为 Genrich 设置-j 参数后的峰的变化，example.bam 文件的两个峰向着外侧移动了。(-y 参数是保留非配对的 align 片段。)

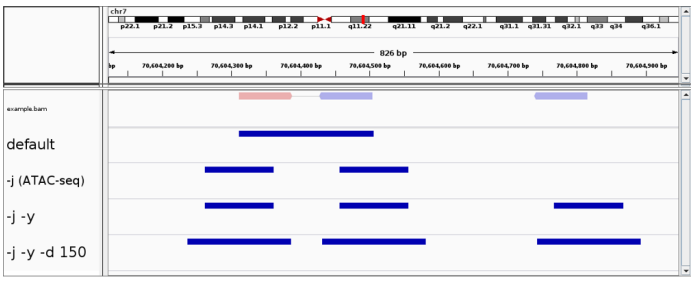


Fig. 3 Genrich 的-j 参数

2 Data Processing in the article

2.1 Methods in the article

The promoters of CLAVATA3 (CLV3) and Rubisco small subunit 2B (RBC) genes were used to drive the expression of the nuclear-targeting

fusion (NTF) gene **in stem cells** and **mesophyll cells** respectively. The transgenic plants CLV3p::NTF;ACT2p::BirA and RBCp::NTF;ACT2p::BirA were constructed.

1. Sequence read mapping and processing

   - Sequencing reads were mapped to the Arabidopsis genome (version TAIR10) using **Bowtie2** software **with default parameters**. The mapped reads in.sam format were converted to.bam format and sorted.
   - **Samtools** was used to filter the reads, **retaining only those with a mapping quality score of 2 or higher** and **mapping to nuclear chromosomes**, **removing reads mapping to the chloroplast or mitochondrial genomes.** The number of mapped reads in **biological replicates** was also made the same for further analysis.

2. Peak calling (detection of transposase hypersensitive sites, THSs) The "Findpeaks" function of the **HOMER** package was used to detect THSs **with the parameters "-minDist 150" and "-region"** . An additional parameter "-regionRes 1" was used when comparing the accessibility between cell types to increase the resolution and number of detected THSs.

3. Genomic distribution analysis of THSs The PAVIS web tool was used to determine the distribution of THSs relative to genomic features, with the "upstream" region set as 2000 bp upstream of the annotated transcription start site (TSS) and the "downstream" region set as 1000 bp downstream of the transcript end site.

4. Identification of THSs enriched in a specific cell type The HTSeq' s htseq-count script was used to obtain the number of reads (counts) at each THS in the stem cell and mesophyll ATAC-seq data for each cell type. The counts were processed with DESeq2, and THSs with an adjusted and a log fold change of 1 or more in a specific cell type were identified as THSs enriched in that cell type

2.2 文章内容

文章通过研究拟南芥茎间分生组织与分化的叶肉细胞的染色质可及性，研究染色质可及性与细胞命运的关系。第一，干细胞和叶肉细胞的染色质可及性特征在定性水平上非常相似，但也发现了数千个在定量上具有不同染色质可接近性的区域。叶肉细胞中优先可达的染色质区域在干细胞中也基本上可达，而反之则不成立。这也说明了干细胞分化过程中基因往往处于一种就绪的状态，染色质可及性高；分化后可及性降低。

3 Data Processing

以下所有的图，从上到下，从左到右：为从 stem cell 到叶肉细胞。

3.1 Data source and Environment Preparation

```
1     conda create -n atac
2     conda activate -n atac
3     conda install anaconda::python-3.7
4     conda install -c bioconda bowtie2
5     conda install -c bioconda samtools
6     conda install -c bioconda deeptools
7     conda install -c bioconda bedtools
8     conda install bioconda::sambamba
```

3.2 Data fetch

Search the GEO ID in ENA to get the link of the SRR files.Use 'wget <URL> outdir ' to download.

3.3 Quality-Control

```
1
2  #!/bin/sh
3  #PBS -N QC
```

```
#PBS -o /home/bioinfo2/PB22071455/bioinfo2024/atac/log/qc.log
#PBS -e /home/bioinfo2/PB22071455/bioinfo2024/atac/log/qc.err
#PBS -q batch
#PBS -l nodes=1:ppn=1
#PBS -l walltime=12:00:00

# 激活conda环境
source ~/.bashrc
cd /home/bioinfo2/PB22071455/bioinfo2024/atac
conda activate atac

less /home/bioinfo2/PB22071455/bioinfo2024/atac/data/data.txt |while read id;
do
fastp -i ./data/${id}_1.fastq.gz -o \
 ./result/fastqc/${id}_1_clean.fastq.gz\
 -I ./data/${id}_2.fastq.gz -O \
 ./result/fastqc/${id}_2_clean.fastq.gz
done

less /home/bioinfo2/PB22071455/bioinfo2024/atac/data/data.txt |while read id;
do
fastp -i ./data/${id}_1.fastq.gz -o ./result/fastqc/${id}_1_cut.fastq.gz\
 -I ./data/${id}_2.fastq.gz -O ./result/fastqc/${id}_2_cut.fastq.gz\
 -f 6 -t 1 -L
done

#做完质控之后再fastqc
```

```
31
32 less /home/bioinfo2/PB22071455/bioinfo2024/atac/data/data.
       txt |while read id;
33 do
34 fastqc ./data/${id}_1.fastq.gz -o ./result/fastqc
35 fastqc ./data/${id}_2.fastq.gz -o ./result/fastqc
36 fastqc ./result/fastqc/${id}_2_clean.fastq.gz -o ./result/
       fastqc
37 fastqc ./result/fastqc/${id}_1_clean.fastq.gz -o ./result/
       fastqc
38 fastqc ./result/fastqc/${id}_2_cut.fastq.gz -o ./result/
       fastqc
39 fastqc ./result/fastqc/${id}_1_cut.fastq.gz -o ./result/
       fastqc
40 done
```

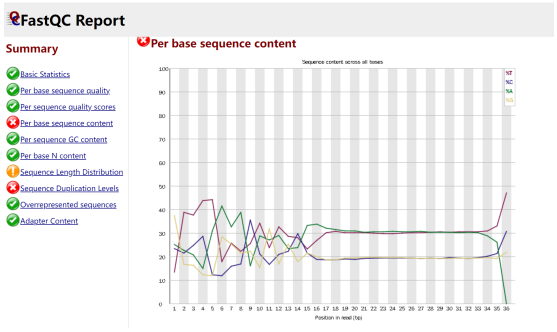


Fig. 4 rawdata QC

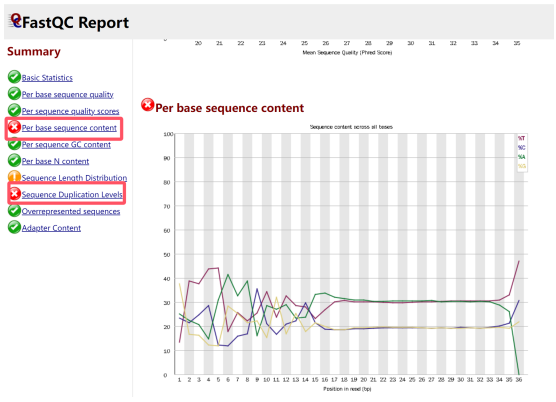


Fig. 5 默认参数过滤 QC

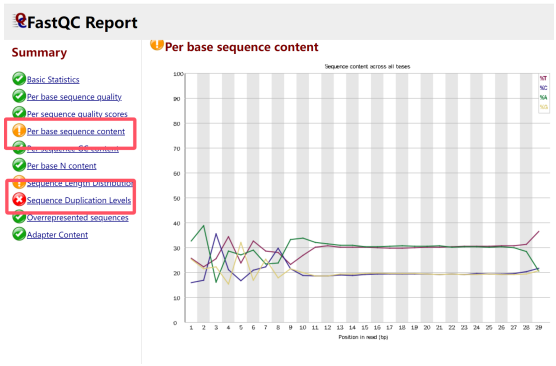


Fig. 6 剪切碱基后 QC 报告

fastqc 解读：

Per base sequence quality： Q 值通过测序 Phred 值计算而得，公式为：Q-score = -10 lg P，其中 Q20 为每 100 个碱基中会有一个识别错，即正确识别率为 99%。

Per Base Sequence Content：由于测序平台及测序长度不同，以及测序仪开始状态不稳定经常出现前后波动情况。

Sequence Duplication Levels：统计序列完全一致的 reads 的频率，横坐标是 duplication 的次数，纵坐标是 duplicated reads 的数目。由于 PCR 过程，可能出现该位置报错（进行 ATAC-seq 时存在该问题）。该处报错无需在意，因为后续会进行 PCR 去重等操作。

Adapter Content: 接头

质控报告显示主要存在'Per Base Sequence Content' 以及'Sequence Duplication Levels' 报错。对于报错'Sequence Duplication Levels'，是由于ATAC-seq 过程中进行了 PCR，后续会进行 PCR 去重。

对于第一个报错，可能是由于测序平台不稳定导致。通过切除前面 8 个和总体趋势完全不同的碱基（剪掉 16 个后面 align 可能跑不了，虽然 PCR 的引物一般也是十几个 bp。）

3.4 Alignment

3.4.1 bwa

Run the alignment ratio pbs(view.pbs) to get the report of alignment.

```
#!/bin/sh

#PBS -N align
#PBS -o /home/bioinfo2/PB22071455/bioinfo2024/atac/log/
    align.log
#PBS -e /home/bioinfo2/PB22071455/bioinfo2024/atac/log/
    align.err
#PBS -q batch
#PBS -l nodes=1:ppn=1
#PBS -l walltime=24:00:00

# 激活conda环境
source ~/.bashrc
cd /home/bioinfo2/PB22071455/bioinfo2024/atac
conda activate atac

bwa index -a bwtsw /home/bioinfo2/PB22071455/bioinfo2024/
    atac/index/Arabidopsis_thaliana.TAIR10.dna.toplevel.fa

#生成sam

less /home/bioinfo2/PB22071455/bioinfo2024/atac/data/data.
```

```
    txt |while read id;
do
#cleandata

bwa mem -v 3 -t 4 \
  /home/bioinfo2/PB22071455/bioinfo2024/atac/index/
      Arabidopsis_thaliana.TAIR10.dna.toplevel.fa \
 ./result/fastqc/${id}_1_clean.fastq.gz ./result/fastqc/${
    id}_2_clean.fastq.gz\
 -o ./result/bwa_sam/${id}_clean_bwa.sam

#cutdata

bwa mem -v 3 -t 4 \
  /home/bioinfo2/PB22071455/bioinfo2024/atac/index/
      Arabidopsis_thaliana.TAIR10.dna.toplevel.fa \
 ./result/fastqc/${id}_1_cut.fastq.gz ./result/fastqc/${id
    }_2_cut.fastq.gz\
 -o ./result/bwa_sam/${id}_cut_bwa.sam

done

#sam-bam
less /home/bioinfo2/PB22071455/bioinfo2024/atac/data/data.
    txt |while read id;
do
#cleandata

samtools view -@ 4 ./result/bwa_sam/${id}_clean_bwa.sam -
    bF 12 -q 10 -O bam\
 -o ./result/bwa_bam_samtools10/${id}_clean_samtools10.bam
#cutdata

```

```
samtools view -@ 4 ./result/bwa_sam/${id}_cut_bwa.sam -bF
    12 -q 10 -O bam\
 -o ./result/bwa_bam_samtools10/${id}_cut_samtools10.bam

done
```

Use the default parameters of bwa to conduct the process.

For **SRR5874657_clean**

```
83161602 + 0 in total (QC-passed reads + QC-failed reads)
83161602 + 0 primary
0 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
82780352 + 0 mapped (99.54% : N/A)
82780352 + 0 primary mapped (99.54% : N/A)
83161602 + 0 paired in sequencing
41580801 + 0 read1
41580801 + 0 read2
82453508 + 0 properly paired (99.15% : N/A)
82661296 + 0 with itself and mate mapped
119056 + 0 singletons (0.14% : N/A)
14606 + 0 with mate mapped to a different chr
3826 + 0 with mate mapped to a different chr (mapQ>=5)
```

解读:

83161602 + 0 in total (QC-passed reads + QC-failed reads)

This indicates that a total of 83,161,602 reads were processed, all of which passed quality control (QC). There are no QC-failed reads (i.e., all reads are of good quality).

83161602 + 0 primary

This shows that 83,161,602 reads have been aligned as "primary" alignments. Primary alignments are the best alignments for each read (the most confident mapping).

0 + 0 secondary

There are no ”secondary” alignments. Secondary alignments occur when a read maps to multiple locations, and the best alignment is considered primary, while the others are secondary.

0 + 0 supplementary

There are no ”supplementary” alignments. Supplementary alignments are used for reads that are split into multiple parts (e.g., spliced alignments for RNA-seq) and need more than one alignment record. 0 + 0 duplicates

No duplicate reads were found. Duplicates can arise during sequencing or sample preparation, where the same fragment is read multiple times.

0 + 0 primary duplicates

There are no primary duplicates, meaning there are no high-confidence duplicate reads in the dataset. 82780352 + 0 mapped (99.54% : N/A)

A total of 82,780,352 reads (99.54% of the total) successfully mapped to the reference genome. This indicates a high alignment rate.

82780352 + 0 primary mapped (99.54% : N/A)

All 82,780,352 reads that were mapped are primary alignments, with the same 99.54% mapping rate.

83161602 + 0 paired in sequencing

All 83,161,602 reads are paired-end reads, meaning each read comes from one end of a DNA fragment, and its pair comes from the other end.

41580801 + 0 read1

Of the paired-end reads, 41,580,801 are from the first read of each pair.

41580801 + 0 read2

The other 41,580,801 are from the second read of each pair.

82453508 + 0 properly paired (99.15% : N/A)

82,453,508 read pairs (99.15%) are considered ”properly paired,” meaning both ends of the read map to the reference genome and are in the correct orientation.

82661296 + 0 with itself and mate mapped

82,661,296 read pairs had both ends successfully mapped to the reference genome, showing both reads in a pair are aligned.

119056 + 0 singletons (0.14% : N/A)

119,056 reads are singletons, meaning only one of the paired reads mapped to the reference genome while the other did not. This is 0.14% of the total reads.

14606 + 0 with mate mapped to a different chr

14,606 read pairs have one read mapped to one chromosome, and the other read mapped to a different chromosome.

3826 + 0 with mate mapped to a different chr (mapQ>=5)

3,826 read pairs have one read mapped to a different chromosome, and the other also maps to a different chromosome, but the mapping quality (MAPQ) of these alignments is greater than or equal to 5. MAPQ is a score that indicates the confidence of the alignment (higher is better).

3.4.2 bowtie2

```
1   #!/bin/sh
2
3  #PBS -N align
4  #PBS -o /home/bioinfo2/PB22071455/bioinfo2024/atac/logs/
       align.log
5  #PBS -e /home/bioinfo2/PB22071455/bioinfo2024/atac/logs/
       align.err
6  #PBS -q batch
7  #PBS -l nodes=1:ppn=4 # 设置为4个线程
8  #PBS -l walltime=24:00:00
9
10 # 加载所需的模块
11
12 # 激活conda环境
13 source ~/.bashrc
14 cd /home/bioinfo2/PB22071455/bioinfo2024/atac
15 conda activate atac
16
17 # 创建索引
```

```
bowtie2-build /home/bioinfo2/PB22071455/bioinfo2024/atac/bowtie2_index/Arabidopsis_thaliana.TAIR10.dna.toplevel.fa \
 /home/bioinfo2/PB22071455/bioinfo2024/atac/bowtie2_index/TAIR10_index

# 比对 (使用4个线程)
while read id; do
    bowtie2 -x /home/bioinfo2/PB22071455/bioinfo2024/atac/bowtie2_index/TAIR10_index \
        -1 ./result/fastqc/${id}_1_clean.fastq.gz \
        -2 ./result/fastqc/${id}_2_clean.fastq.gz \
        -S ./results/${id}.sam -p 4
done < /home/bioinfo2/PB22071455/bioinfo2024/atac/data/data.txt

while read id; do
#sam转bam, 质量大于2
samtools view -@ 4 ./results/${id}.sam -bF 12 -q 2 -O bam -o ./results/${id}_samtools10.bam
#sort
sambamba sort -t 2 \
 -o ./results/${id}_sorted.bam ./results/${id}_samtools10.bam

#PCR去重
sambamba markdup -r -t 4 ./results/${id}_sorted.bam ./results/${id}_samtools10_rdup.bam

#过滤得到核基因
samtools view -@ 4 -b ./results/${id}_samtools10_rdup.bam 1 2 3 4 5 > ./results/peakcall/${id}_samtools10_rdup_chr.bam
```

```
41
42 done < /home/bioinfo2/PB22071455/bioinfo2024/atac/data/
       data.txt
```

bowtie2 结果：

```
41580801 reads; of these:
 41580801 (100.00%) were paired; of these:
  405333 (0.97%) aligned concordantly 0 times
  26927499 (64.76%)   aligned concordantly exactly 1 time
 14247969 (34.27%) aligned concordantly >1 times
  ----
  405333 pairs aligned concordantly 0 times; of these:
  52170 (12.87%) aligned discordantly 1 time
  ----
  353163 pairs aligned 0 times concordantly or discordantly; of these:
  706326 mates make up the pairs; of these:
    524458 (74.25%) aligned 0 times
    66967 (9.48%) aligned exactly 1 time
    114901 (16.27%) aligned >1 times
99.37% overall alignment rate
```

3.5 Filter1

```
1     #!/bin/sh
2  #PBS -N 1filter
3  #PBS -o /home/bioinfo2/PB22071455/bioinfo2024/atac/log/1
       filter.log
4  #PBS -e /home/bioinfo2/PB22071455/bioinfo2024/atac/log/1
       filter.err
5  #PBS -q batch
6  #PBS -l nodes=1:ppn=1
7  #PBS -l walltime=24:00:00
```

```
8
9  # 加载所需的模块
10
11 # 激活conda环境
12 source ~/.bashrc
13 cd /home/bioinfo2/PB22071455/bioinfo2024/atac/result/
       bwa_bam_samtools10/
14 conda activate atac
15
16
17 #去重
18 less /home/bioinfo2/PB22071455/bioinfo2024/atac/data/data.
       txt |while read id;
19 do
20 #cleandata
21 sambamba markdup -r -t 4 ${id}_clean_samtools10.bam\
22  ${id}_clean_samtools10_rdup.bam
23
24 #cundata
25 sambamba markdup -r -t 4 ${id}_cut_samtools10.bam\
26  ${id}_cut_samtools10_rdup.bam
27 done
28
29 #可视化
30 less /home/bioinfo2/PB22071455/bioinfo2024/atac/data/data.
       txt |while read id;
31 do
32 #cleandata
33
34 #bam-bw
35 ##将bam文件 排序 -@ 4 使用四个线程
36 samtools sort -@ 4 -O bam \
37  -o ${id}_clean_samtools10_rdup_sort.bam\
```

```
 -T ${id}_clean_samtools10.temp\
 ${id}_clean_samtools10_rdup.bam
##对bam文件建立index
samtools index -@ 4 \
 ${id}_clean_samtools10_rdup_sort.bam
##然后生成bw文件 -t 4 使用四个线程
BAMscale scale -t 4 --bam\
 ${id}_clean_samtools10_rdup_sort.bam

#cutdata
##将bam文件 排序 -@ 4 使用四个线程
samtools sort -@ 4 -O bam \
 -o ${id}_cut_samtools10_rdup_sort.bam\
 -T ${id}_cut_samtools10.temp\
 ${id}_cut_samtools10_rdup.bam
##对bam文件建立index
samtools index -@ 4 \
 ${id}_cut_samtools10_rdup_sort.bam
##然后生成bw文件 -t 4 使用四个线程
BAMscale scale -t 4 --bam\
 ${id}_cut_samtools10_rdup_sort.bam

done
```

3.6 Visualization(IGV)

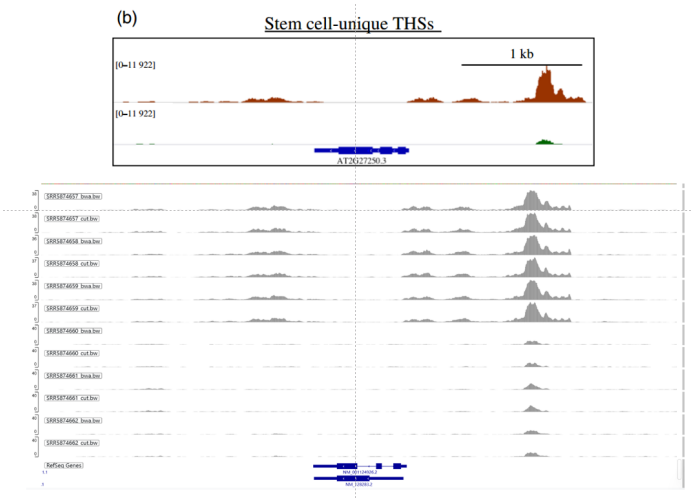


Fig. 7 干细胞特异的 peak

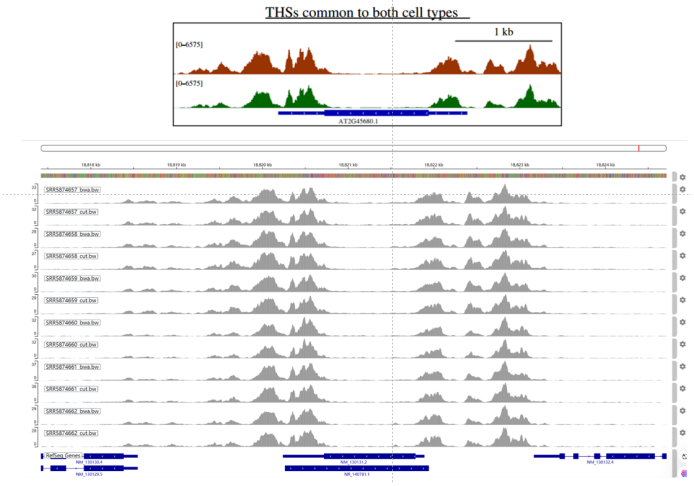


Fig. 8 共有的 peak

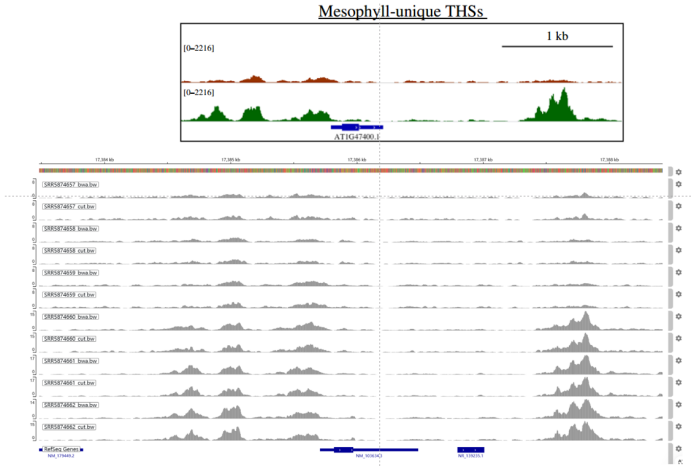


Fig. 9 叶肉细胞 peak

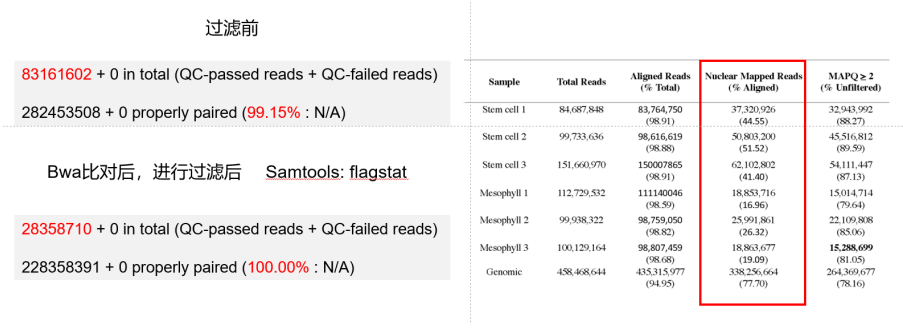
有些位点特异性的在叶肉细胞在开放，有些位点在干细胞中特异的开放。这种基因组的动态变化揭示分化的过程。

3.7 Datapre for peakcall

```
1   #!/bin/sh
2
3  #PBS -N callpre
4  #PBS -o /home/bioinfo2/PB22071455/bioinfo2024/atac/log/
       callpre.log
5  #PBS -e /home/bioinfo2/PB22071455/bioinfo2024/atac/log/
       callpre.err
6  #PBS -q batch
7  #PBS -l nodes=1:ppn=1
8  #PBS -l walltime=12:00:00
9
10 # 加载所需的模块
11
12 # 激活conda环境
13 source ~/.bashrc
```

```
cd /home/bioinfo2/PB22071455/bioinfo2024/atac
conda activate atac

#去除细胞器基因组
less /home/bioinfo2/PB22071455/bioinfo2024/atac/data/data.txt |while read id;
do
#cleandata
samtools view -@ 4 -b \
 ./result/bwa_bam_samtools10/${id}_clean_samtools10_rdup_sort.bam 1 2 3 4 5 > ./result/peakcall/${id}_clean_samtools10_rdup_chr.bam

#cutdata
samtools view -@ 4 -b \
 ./result/bwa_bam_samtools10/${id}_cut_samtools10_rdup_sort.bam 1 2 3 4 5 > ./result/peakcall/${id}_cut_samtools10_rdup_chr.bam
done


#sort

less /home/bioinfo2/PB22071455/bioinfo2024/atac/data/data.txt |while read id;
do
#cleandata
samtools sort -n -@ 4 -O bam\
 -o ./result/peakcall/${id}_clean_samtools10_rdup_chr_sortn.bam\
 -T ${id}_clean_samtools10.temp \
 ./result/peakcall/${id}_clean_samtools10_rdup_chr.bam

```

```
#cutdata

samtools sort -n -@ 4 -O bam\
 -o ./result/peakcall/${id}_cut_samtools10_rdup_chr_sortn.bam\
 -T ${id}_cut_samtools10.temp \
 ./result/peakcall/${id}_cut_samtools10_rdup_chr.bam
done
```



Sample	Total Reads	Aligned Reads (% Total)	Nuclear Mapped Reads (% Aligned)	MAPQ ≥ 2 (% Unfiltered)
Stem cell 1	84,687,848	83,764,750 (98.91)	37,320,926 (44.55)	32,943,992 (88.27)
Stem cell 2	99,733,636	98,616,619 (98.88)	50,803,200 (51.52)	45,516,812 (89.59)
Stem cell 3	151,660,970	150007865 (98.91)	62,102,802 (41.40)	54,111,447 (87.13)
Mesophyll 1	112,729,532	111140046 (98.59)	18,853,716 (16.96)	15,014,714 (79.64)
Mesophyll 2	99,938,322	98,759,050 (98.82)	25,991,861 (26.32)	22,109,808 (85.06)
Mesophyll 3	100,129,164	98,807,459 (98.68)	18,863,677 (19.09)	**15,288,699** (81.05)
Genomic	458,468,644	435,315,977 (94.95)	338,256,664 (77.70)	264,369,677 (78.16)

Fig. 10 过滤前后

过滤后的结果发现很多仅有 30% mapping 到基因组上。也是和论文很符合。

3.8 Peakcall

3.8.1 Genrich

```
    #!/bin/sh
#PBS -o /home/bioinfo2/PB22071455/bioinfo2024/atac/log/2peakcall.log
#PBS -e /home/bioinfo2/PB22071455/bioinfo2024/atac/log/2peakcall.err
#PBS -q batch
#PBS -l nodes=1:ppn=1
#PBS -l walltime=12:00:00
```

```
7
8  # 加载所需的模块
9
10 # 激活conda环境
11 source ~/.bashrc
12 cd /home/bioinfo2/PB22071455/bioinfo2024/atac/result/
       peakcall
13 conda activate atac
14
15
16 less /home/bioinfo2/PB22071455/bioinfo2024/atac/data/data.
       txt |while read id;
17 do
18
19 Genrich -t ${id}_cut_samtools10_rdup_chr_sortn.bam -o ${id
       }_cut2.narrowPeak -j -v
20 Genrich -t ${id}_clean_samtools10_rdup_chr_sortn.bam -o ${
       id}_clean2.narrowPeak -j -v
21
22 done
23
24 Genrich -t SRR5874660_cut_samtools10_rdup_chr_sortn.bam,
       SRR5874661_cut_samtools10_rdup_chr_sortn.bam,
       SRR5874662_cut_samtools10_rdup_chr_sortn.bam \
25  -o leaf_2cut.narrowPeak -j -v
26
27 Genrich -t SRR5874660_clean_samtools10_rdup_chr_sortn.bam,
       SRR5874661_clean_samtools10_rdup_chr_sortn.bam,
       SRR5874662_clean_samtools10_rdup_chr_sortn.bam \
28  -o leaf_2clean.narrowPeak -j -v
29
30 Genrich -t SRR5874657_clean_samtools10_rdup_chr_sortn.bam,
       SRR5874658_clean_samtools10_rdup_chr_sortn.bam,
```

```
    SRR5874659_clean_samtools10_rdup_chr_sortn.bam \
 -o stem_2clean.narrowPeak -j -v

Genrich -t SRR5874657_cut_samtools10_rdup_chr_sortn.bam,
    SRR5874658_cut_samtools10_rdup_chr_sortn.bam,
    SRR5874659_cut_samtools10_rdup_chr_sortn.bam \
 -o stem_2cut.narrowPeak -j -v
```

-j 参数为 Genrich 进行 ATAC-seq 模式的参数。

3.8.2 MACS2 peakcall

Homer 'findpeaks' is used by the article with parameters ' **"-minDist 150" and "-region"** '. However, **the output lacks the p-value and q-value and significance that the '*.narrowPeak' files have(R packages 'ChIPseeker' can not be conducted without the '*.narrowPeak' files).** So after using the Homer 'findpeaks', I use the MACS2 to conduct the peakcall with **similar** parameters.

MACS2 parameters:

-t：指定输入的 BAM 文件。

-f BAM：指定输入文件格式为 BAM。

-g 125000000：此参数表示基因组大小。对于拟南芥（Arabidopsis），参考基因组大小大约是 125,000,000 个碱基对。

–min-length 150：与 HOMER 中的 -minDist 150 类似，设置最小的峰长度为 150 bp。

–keep-dup all：保留所有重复的读数（通常在 ATAC-seq 数据中，重复读数也非常重要）。

–outdir ./results/macs2：指定输出目录。

```
#!/bin/sh

#PBS -N macs2_peakcall
#PBS -o /home/bioinfo2/PB22071455/bioinfo2024/atac/logs/
    macs2.log
```

```
#PBS -e /home/bioinfo2/PB22071455/bioinfo2024/atac/logs/macs2.err
#PBS -q batch
#PBS -l nodes=1:ppn=4
#PBS -l walltime=24:00:00
# 激活conda环境
source ~/.bashrc
cd /home/bioinfo2/PB22071455/bioinfo2024/atac
conda activate atac

# Peak Calling 使用 MACS2
while read id; do
    # 使用 MACS2 进行 Peak Calling, 输出到 ./results/macs2 目录
    macs2 callpeak -t ./results/peakcall/${id}_samtools10_rdup_chr.bam \
                -f BAM \
                -g 125000000 \
                -n ${id}_peaks \
                --nomodel --shift 37 --extsize 7 \
                --min-length 150 --keep-dup all \
                --outdir ./results/macs2
done < /home/bioinfo2/PB22071455/bioinfo2024/atac/data/data.txt
```

3.8.3 Homer peakcall

Output:

.narrowPeak:

```

1 2398 2858 SRR5874657\_peaks\_peak\_1 85 . 3.50024 9.92767 8.55716 238
1 2922 3263 SRR5874657\_peaks\_peak\_2 307 . 6.68227
```

```
    32.5758 30.7821 167
```

homer.txt:

```
1    # HOMER Peaks
2  # Peak finding parameters:
3  # tag directory = ./results/tagdir/SRR5874657_tagDir
4  #
5  # total peaks = 24185
6  # peak size = 136
7  # peaks found using tags on both strands
8  # minimum distance between peaks = 150
9  # fragment length = 70
10 # genome size = 119145877
11 # Total tags = 14393670.5
12 # Total tags in peaks = 4697693.0
13 # Approximate IP efficiency = 32.64%
14 # tags per bp = 0.120367
15 # expected tags per peak = 16.370
16 # maximum tags considered per bp = 12.0
17 # effective number of tags used for normalization =
     10000000.0
18 # Individual peaks have been stitched together into
     variable length regions
19 # Peaks have been centered at maximum tag pile-up
20 # FDR rate threshold = 0.001000000
21 # FDR effective poisson threshold = 1.874920e-05
22 # FDR tag threshold = 36.0
23 # number of putative peaks = 76151
24 #
25 # size of region used for local filtering = 10000
26 # Fold over local region required = 4.00
27 # Poisson p-value over local region required = 1.00e-04
28 # Putative peaks filtered by local signal = 44605
```

```
#
# Maximum fold under expected unique positions for tags = 2.00
# Putative peaks filtered for being too clonal = 3
#
# cmd = findPeaks ./results/tagdir/SRR5874657_tagDir -minDist 150 -region -regionRes 1 -o ./results/homer/SRR5874657_homer_peak.txt -style factor
#
# Column Headers:
#PeakID chr start end strand Normalized Tag Count region size findPeaks Score Fold Change vs Local p-value vs Local Clonal Fold Change
3-3 3 14201502 14201776 + 1622.2 0.611 829.000000 5.65 0.00e+00 0.79
5-131 5 14913711 14914275 + 1564.6 0.550 364.500000 5.79 6.26e-162 0.81
2-45 2 3377995 3378405 + 1341.6 0.554 490.500000 6.76 1.36e-245 0.82
```

So the homer.txt should be converted into .narrowPeak files for further analysis.

转换 homer.txt 文件，homer 本身可以转化为.bed 文件。

```
pos2bed.pl peakfile.txt > peakfile.bed
bed2pos.pl peakfile.bed > peakfile.txt
```

.bed 文件可以用 bedtools 取交集。

不过我没有这样做。

用 chatGPT 帮我写 R 语言直接转化，我没细看，不过后面需要用.narrowPeak 做 ChIPseeeker 分析以及 intervene 的 venn 图，只需要峰位置正确就行了。

```
# Load required library
library(dplyr)

```

```
4  # Function to convert HOMER file to .narrowPeak format
5  homer_to_narrowpeak <- function(homer_file,
       narrowpeak_file) {
6
7    # Step 1: Read the HOMER file, skipping comment lines (
         those starting with '#')
8    homer_data <- read.delim(homer_file, header = FALSE,
         comment.char = "#", stringsAsFactors = FALSE)
9
10   # Step 2: Assign column names to HOMER data
11   colnames(homer_data) <- c("PeakID", "chr", "start", "end
         ", "strand", "Normalized_Tag_Count",
12                             "region_size", "findPeaks_Score",
                                   "Fold_Change_vs_Local", "
                                   p_value_vs_Local",
13                             "Clonal_Fold_Change")
14
15   # Step 3: Convert p_value_vs_Local into numeric and
         calculate additional columns
16   homer_data$p_value_vs_Local <- as.numeric(
         homer_data$p_value_vs_Local)
17
18   # Handle missing or non-numeric p-values
19   homer_data <- homer_data %>%
20     mutate(
21       # Calculate 'score' as -10 * log10(qvalue), using p-
             value as qvalue here
22       score = ifelse(!is.na(p_value_vs_Local) &
             p_value_vs_Local > 0,
23                      -10 * log10(p_value_vs_Local),
24                      NA), # Assign NA if p-value is invalid
                            or missing
25
```

```
26      # signalValue is typically fold_enrichment (Fold
            Change vs Local)
27      signalValue = as.numeric(Fold_Change_vs_Local),
28
29      # Calculate p-value as -log10(p-value)
30      pValue = ifelse(!is.na(p_value_vs_Local) &
            p_value_vs_Local > 0,
31                      -log10(p_value_vs_Local),
32                      NA),
33
34      # qValue (can be same as p-value for simplicity)
35      qValue = pValue, # We assume q-value as -log10(p-
            value)
36
37      # peak: Calculate center of the peak (summit is
            assumed to be 0)
38      peak = (start + end) / 2 - start
39    )
40
41  # Step 4: Convert start to 0-based by subtracting 1
42  homer_data$start <- homer_data$start - 1
43
44  # Step 5: Prepare the data in the .narrowPeak format
45  narrowpeak_data <- homer_data %>%
46    select(chr, start, end, PeakID, score, strand,
          signalValue, pValue, qValue, peak)
47
48  # Step 6: Write the output to the .narrowPeak file
49  write.table(narrowpeak_data, narrowpeak_file, sep = "\t
      ", quote = FALSE, row.names = FALSE, col.names =
      FALSE)
50
51  # Print success message
```

```
52   print(paste("Conversion successful! The .narrowPeak file
         is saved at", narrowpeak_file))
53 }
54
55 # Example file paths
56 homer_file <- '/home/bioinfo2/PB22071455/bioinfo2024/atac/
       results/homer/SRR5874657_homer_peak.txt'
57 narrowpeak_file <- '/home/bioinfo2/PB22071455/bioinfo2024/
       atac/results/homer/SRR5874657_homer_peak.narrowPeak'
58
59 # Run the conversion function
60 homer_to_narrowpeak(homer_file, narrowpeak_file)
61
62 # Another example with a different file
63 homer_file <- '/home/bioinfo2/PB22071455/bioinfo2024/atac/
       results/homer/SRR5874658_homer_peak.txt'
64 narrowpeak_file <- '/home/bioinfo2/PB22071455/bioinfo2024/
       atac/results/homer/SRR5874658_homer_peak.narrowPeak'
65
66 # Run the conversion function
67 homer_to_narrowpeak(homer_file, narrowpeak_file)
68
69 # One more example with a third file
70 homer_file <- '/home/bioinfo2/PB22071455/bioinfo2024/atac/
       results/homer/SRR5874659_homer_peak.txt'
71 narrowpeak_file <- '/home/bioinfo2/PB22071455/bioinfo2024/
       atac/results/homer/SRR5874659_homer_peak.narrowPeak'
72
73 # Run the conversion function
74 homer_to_narrowpeak(homer_file, narrowpeak_file)
```

只能说 homer 不是很好用呀，直接生成的.bed 文件我也没仔细了解是不是可以直接用。用这个.bed 文件和论文给的.bed 文件做 intervene 的 venn 图，结果交集是 0，我就没用这个.bed 文件了。直接道心破碎。懒得探索

homer 了，伤心。

4 Venns

```
#!/bin/sh
#PBS -N vene
#PBS -o /home/bioinfo2/PB22071455/bioinfo2024/atac/logs/venn.log
#PBS -e /home/bioinfo2/PB22071455/bioinfo2024/atac/logs/venn.err
#PBS -q batch
#PBS -l nodes=1:ppn=1
#PBS -l walltime=12:00:00

# 加载所需的模块

# 激活conda环境
source ~/.bashrc
cd /home/bioinfo2/PB22071455/bioinfo2024/atac
conda activate atac

#比较Genrich cut 和 clean的结果
intervene venn -i /home/bioinfo2/PB22071455/bioinfo2024/atac/result/peakcall/stem_2clean.narrowPeak\
 /home/bioinfo2/PB22071455/bioinfo2024/atac/result/peakcall/stem_2cut.narrowPeak\
 /home/bioinfo2/PB22071455/bioinfo2024/atac/result/peakcall/leaf_2clean.narrowPeak\
 /home/bioinfo2/PB22071455/bioinfo2024/atac/result/peakcall/leaf_2cut.narrowPeak\
 --save-overlaps\
 --output ./venn/clean_cut.pdf

```

```
#比较Genrich cut 和 clean的结果
intervene venn -i /home/bioinfo2/PB22071455/bioinfo2024/atac/result/peakcall/stem_2clean.narrowPeak\
 /home/bioinfo2/PB22071455/bioinfo2024/atac/result/peakcall/leaf_2clean.narrowPeak\
 --save-overlaps\
 --output ./venn/stem_leaf.pdf

intervene venn -i /home/bioinfo2/PB22071455/bioinfo2024/atac/result/peakcall/stem_2clean.narrowPeak\
 /home/bioinfo2/PB22071455/bioinfo2024/atac/results/macs2/SRR5874658_peaks_peaks.narrowPeak\
 /home/bioinfo2/PB22071455/bioinfo2024/atac/results/homer/SRR5874658_homer_peak.narrowPeak\
 --save-overlaps\
```

后面做得也比较混乱了，所以路径都奇奇怪怪的。

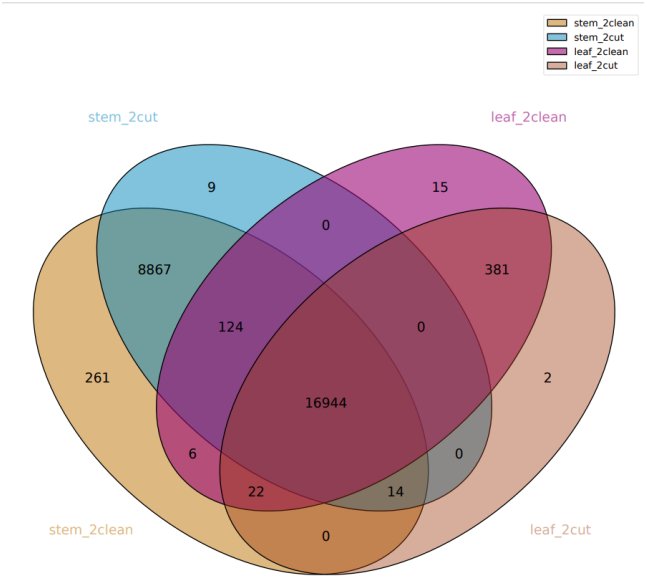


Fig. 11 默认参数过滤以及剪切碱基处理后 Venn 结果

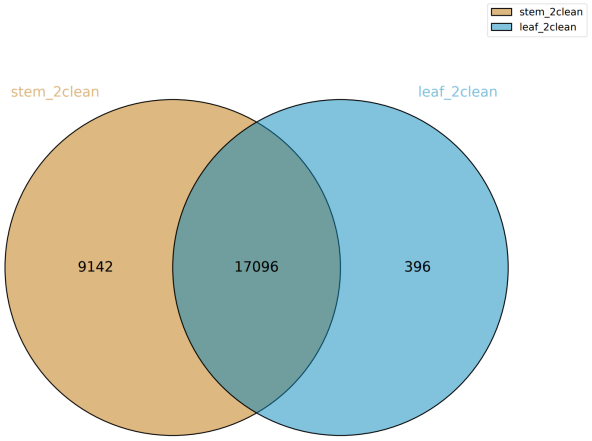


Fig. 12 干细胞以及叶肉细胞 venn 结果

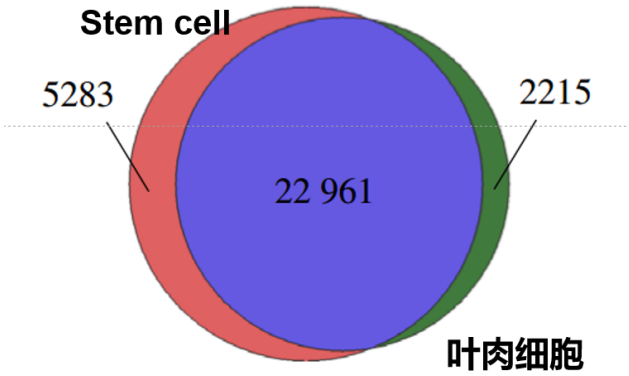


Fig. 13 原论文 venn 图

原论文中叶肉细胞以及干细胞 peak 的 venn 图。

本实验 Genrich 没有设置最小峰间距（原论文 homer 设置了最小峰间距）。可能是叶肉细胞过滤条件太过严格。不同参数可能会导致很大的区别。

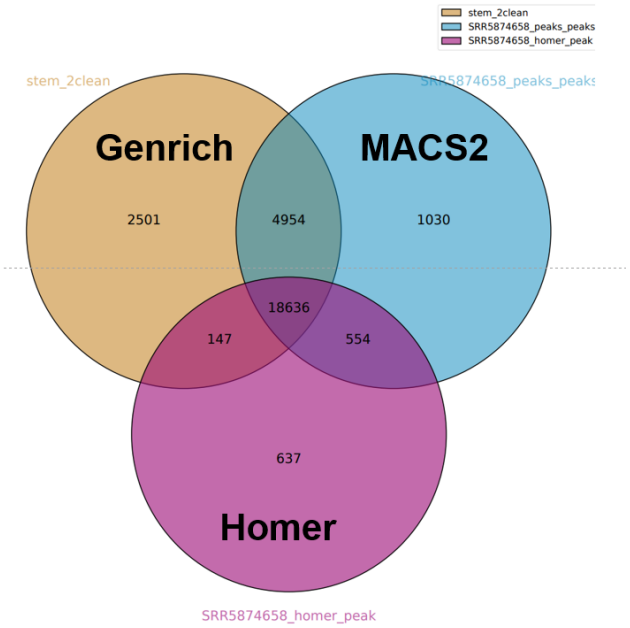


Fig. 14 Genrich,homer,MACS2 结果

三个不同工具，homer 设置了最小峰间距 150bp，因此 peakcall 的结果最少。MACS2 和 Genrich 的结果很接近，仅有 10% 的不同。三者结果的主体是一致的。

5 ChIPseeker

在 jupyter 环境中安装 ChIPseeker 和 Genomicfeatures

```
conda install bioconda::bioconductor-chipseeker
conda install bioconda::bioconductor-genomicfeatures
```

环境配置好后连接 jupyter。

进入 jupyter：

先创建环境然后完成配置

```
conda create -n jupyter jupyter
conda install anaconda::python
```

```
#先创建环境

jupyter notebook --generate-config

jupyter notebook password
#按顺序输入
cd .jupyter
ls
vi jupyter_notebook_config.py
#.+文件为隐藏文件
#最后一行加上一句
c.ServerApp.ip="*"
```

运行 jupyter 脚本

```
 #!/bin/sh
#PBS -N jup
#PBS -o jpt.log
#PBS -e jpt.err
#PBS -q batch
#PBS -l nodes=1:ppn=1
#PBS -l walltime=12:00:00
source ~/.bashrc
cd /home/bioinfo2/PB22071455/
#jupyter网页到达指定的位置
conda activate jupyter

jupyter lab
```

确保该脚本一直在运行后，cat jpt.err 来获取报错，会出现类似下图的报错:

/home/bioinfo2/PB22071455/miniconda3/envs/jupyter/share/jupyter/lab

[I 2024-11-05 16:54:36.567 ServerApp] jupyterlab | extension was successfully loaded.

[I 2024-11-05 16:54:36.760 ServerApp] nbclassic | extension was success-

fully loaded.

[I 2024-11-05 16:54:36.817 ServerApp] The port 8888 is already in use, trying another port.

[I 2024-11-05 16:54:36.828 ServerApp] The port 8889 is already in use, trying another port.

[I 2024-11-05 16:54:36.828 ServerApp] The port 8890 is already in use, trying another port.

[I 2024-11-05 16:54:36.828 ServerApp] The port 8891 is already in use, trying another port.

[I 2024-11-05 16:54:36.829 ServerApp] The port 8892 is already in use, trying another port.

[I 2024-11-05 16:54:36.830 ServerApp] Jupyter Server 1.23.4 is running at:

[I 2024-11-05 16:54:36.830 ServerApp] http://localhost:8979/lab

注意结点为 8979。

用 qstat -nu PB22071455 确定电脑节点位置。

在新的界面输入:

ssh -NL localhost:9000:comput8:8979 PB22071455@nebula.ustc.edu.cn

其中，9000 为自定义的命名，后续需要到浏览器中输入 localhost:9000。

comput8 为节点位置，8979 为报错返回的地址。

输入密码之后会一直挂住。

到浏览器中输入 localhost:9000。

建立 R kernel.

在 jupyter 环境中运行:

```
conda activate jpt
conda install r-base=4.3.3
#输入R进入r-base环境中:
R
IRkernel::installspec(name='ir433',displayname ='R 4.3.3')
```

即可建立一个 R kernel 进行远程服务器连接并运行 R 语言的包。

建立 R 的 kernel 后即可运行下列代码:

下面是做 Peak 分布的 R 语言脚本。

```
1
2  # 加载所需的包
3  library('ChIPseeker')
4  library("GenomicFeatures")
5
6  # 创建 TxDb 对象, 指定 GFF3 文件路径
7  ara_TxDb <- makeTxDbFromGFF("~/bioinfo2024/atac/gff3/
       Arabidopsis_thaliana.TAIR10.51.gff3")
8
9  # 设置工作目录为 MACS2 结果目录
10 setwd("~/bioinfo2024/atac/results/macs2/")
11
12 # 获取该文件夹下所有的 narrowPeak 文件名
13 # 请根据实际情况修改路径, 确保所有需要的 narrowPeak 文件都被
       列出
14 files <- list.files(pattern = "*.narrowPeak")
15 files_list <- as.list(files)
16
17 # 为每个文件指定样本名称
18 names(files_list) <- c(
19   "SRR5874657_peaks", "SRR5874658_peaks", "
         SRR5874659_peaks",
20   "SRR5874660_peaks", "SRR5874661_peaks", "
         SRR5874662_peaks"
21 )
22
23 # 对 peak 进行注释
24 peakAnnoList <- lapply(files_list, annotatePeak, TxDb =
       ara_TxDb, tssRegion = c(-1000, 1000), verbose = FALSE)
25
26 # 开始保存图像为 PNG 文件
27 png(filename = "Distribuation.png", width = 700, height =
       600) # 设置 PNG 输出文件及图像尺寸
```

```
28
29 # 可视化多个样本 peak 和 TSS 之间距离的分布
30 plotDistToTSS(peakAnnoList, ylab = "Opening sites (%)
       (5'->3')", title = paste0("Distribution of opening
       loci relative to TSS"))
31
32 # 关闭设备，保存图像
33 dev.off()
```

下面是做热图的脚本。由于这个图非常的大，分辨率太低会导致出现很多白色的条纹，因此像素才设置的很大。

```
1  # 加载所需的包
2  library('ChIPseeker')
3  library("GenomicFeatures")
4
5  # 创建 TxDb 对象，指定 GFF3 文件路径
6  ara_TxDb <- makeTxDbFromGFF("~/bioinfo2024/atac/gff3/
       Arabidopsis_thaliana.TAIR10.51.gff3")
7
8  # 设置工作目录为 MACS2 结果目录
9  setwd("~/bioinfo2024/atac/results/macs2/")
10
11 # 获取该文件夹下所有的 narrowPeak 文件名
12 # 请根据实际情况修改路径，确保所有需要的 narrowPeak 文件都被
       列出
13 files <- list.files(pattern = "*.narrowPeak")
14 files_list <- as.list(files)
15
16 # 为每个文件指定样本名称
17 names(files_list) <- c(
18   "SRR5874657_peaks", "SRR5874658_peaks", "
         SRR5874659_peaks",
19   "SRR5874660_peaks", "SRR5874661_peaks", "
```

```
        SRR5874662_peaks"
)

# 打开 PNG 图形设备，设置图像分辨率和大小
png(filename = "Heatmap_stem.png", width = 30000, height =
    8000, res = 1000)

# 可视化多个样本 peak 在 1kb 范围内的分布热图
peakHeatmap(files_list, weightCol = "V5", TxDb = ara_TxDb,
            upstream = 1000, downstream = 1000)

# 关闭设备，保存图像
dev.off()
```

绘制 peak 所在位置的饼状图的，这只以其中一份数据作为例子。

```
# 加载所需的包
library('ChIPseeker') # This will now work after
    installing GenomeInfoDbData
library("GenomicFeatures")

# 创建 TxDb 对象，指定 GFF3 文件路径
ara_TxDb <- makeTxDbFromGFF("~/bioinfo2024/atac/gff3/
    Arabidopsis_thaliana.TAIR10.51.gff3")

# 设置工作目录为 MACS2 结果目录
setwd("~/bioinfo2024/atac/results/macs2/")

# 获取该文件夹下所有的 narrowPeak 文件名
# 请根据实际情况修改路径，确保所有需要的 narrowPeak 文件都被
    列出
files <- list.files(pattern = "*.narrowPeak")
files_list <- as.list(files)

```

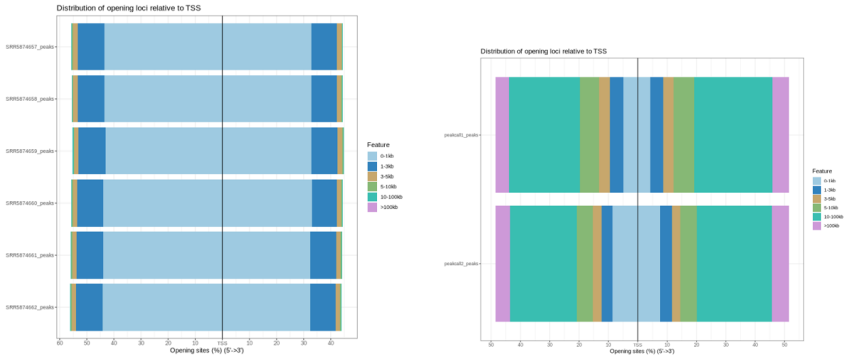
```
# 为每个文件指定样本名称
names(files_list) <- c(
  "SRR5874657_peaks", "SRR5874658_peaks", "
      SRR5874659_peaks",
  "SRR5874660_peaks", "SRR5874661_peaks", "
      SRR5874662_peaks"
)

# 对文件列表中的每个文件进行注释
peakAnnoList <- lapply(files_list, annotatePeak, TxDb =
    ara_TxDb, tssRegion = c(-3000, 3000), verbose = FALSE)

# 选择需要的文件进行注释并生成饼图
peakAnno <- annotatePeak(files[[4]],
                         tssRegion = c(-3000, 3000),
                         TxDb = ara_TxDb,
                         annoDb = "org.At.tair.db") # 使用
                             Arabidopsis的基因注释数据库

# 开始保存图像为PNG文件
png(filename = "Pie_stem.png", width = 800, height = 600)
    # 设置PNG输出文件和图像尺寸

# 生成并保存 annotation pie 图
plotAnnoPie(peakAnno)

# 关闭设备，保存图像
dev.off()
```

绘制 peak 的谱图。

```
# 加载所需的包
library('ChIPseeker')
library("GenomicFeatures")
```

```
4
5  # 创建 TxDb 对象
6  ara_TxDb <- makeTxDbFromGFF("~/bioinfo2024/atac/gff3/
       Arabidopsis_thaliana.TAIR10.51.gff3")
7
8  # 设置工作目录为 MACS2 结果目录
9  setwd("~/bioinfo2024/atac/results/macs2/")
10
11 # 获取该文件夹下所有的 narrowPeak 文件名
12 # 请根据实际情况修改路径，确保所有需要的 narrowPeak 文件都被
       列出
13 files <- list.files(pattern = "*.narrowPeak")
14 files_list <- as.list(files)
15
16 # 为每个文件指定样本名称
17 names(files_list) <- c(
18   "SRR5874657_peaks", "SRR5874658_peaks", "
         SRR5874659_peaks",
19   "SRR5874660_peaks", "SRR5874661_peaks", "
         SRR5874662_peaks")
20
21 # 获取启动子区域，包含1kb的上游和下游区域
22 promoter <- getPromoters(TxDb = ara_TxDb, upstream = 1000,
       downstream = 1000)
23
24 # 为每个文件生成tag矩阵
25 tagMatrixList <- lapply(files_list, getTagMatrix, windows
       = promoter)
26
27 # 创建平均图谱
28 plot_obj <- plotAvgProf(tagMatrixList, xlim = c(-1000,
       1000), conf = 0.95, resample = 500, facet = "row")
29
```

```
# 设定PDF输出文件及图像尺寸
pdf("Frequency.pdf", width = 10, height = 8)

# 绘制平均图谱
plotAvgProf(tagMatrixList, xlim = c(-1000, 1000), conf =
    0.95, resample = 500, facet = "row")

# 关闭设备，保存图像
dev.off()
```



(a) ATAC-seq 的 peak 分布　　(b) ChIP-seq 的 peak 分布

Fig. 15 Main caption for both images

从 ChIPseeker 结果说明，ATAC-seq 揭示出拟南芥的 peak 主要分布在 1kb 范围内。peak 范围非常接近 TSS（转录起始位点）位置，揭示出拟南芥的调控主要是近端调控，即顺式调控因子。大多数处于 TSS 上游。

ChIP-seq 结果，peak 主体分布在 10kb 范围外，下拉 TF，TF 为调节因子，即反式调控元件。

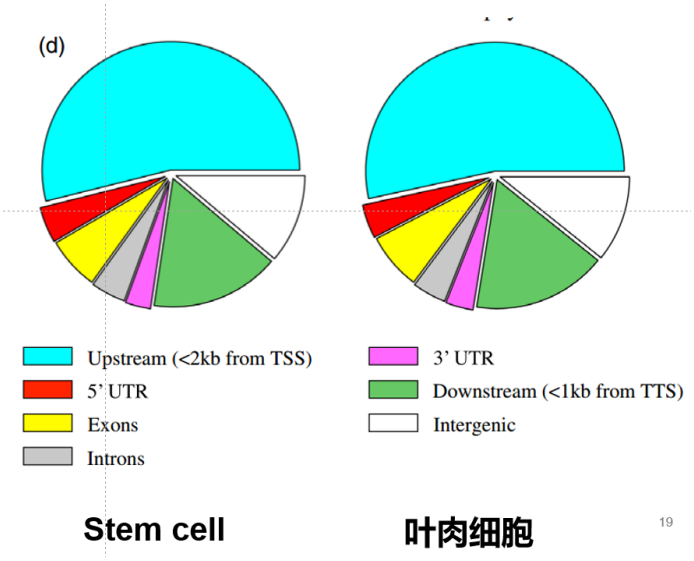


Fig. 16 论文中 peak 分布

论文中分布揭示出 peak 的分布主要是上游，和 ChIPseeker 揭示出来的一致。

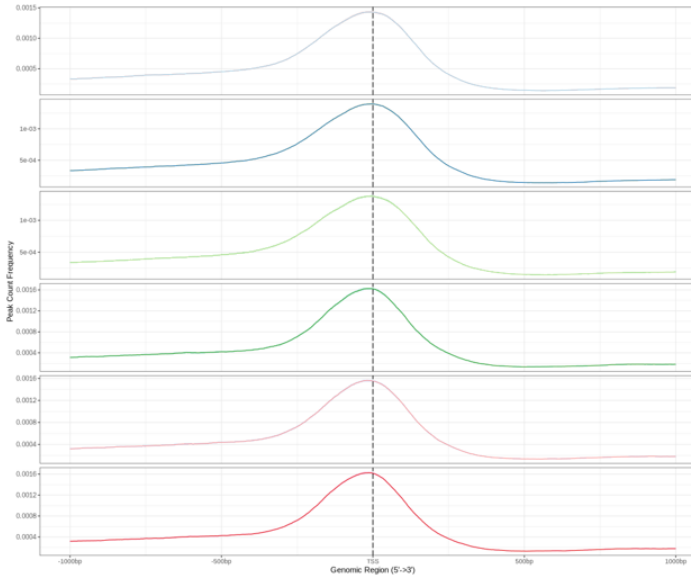


Fig. 17 谱图

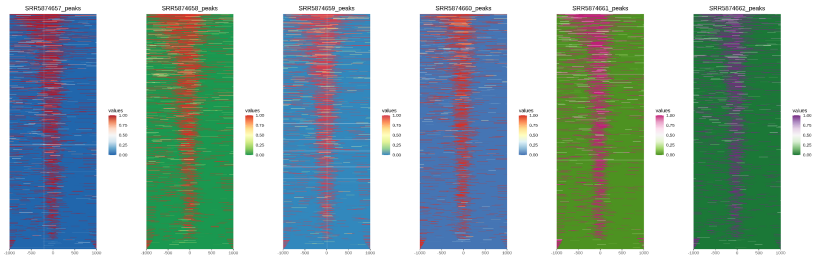


Fig. 18 heatmap

谱图以及热图可以揭示出 stem 细胞峰更宽，开放程度更大。本次实验只做了上游的一些分析。但是还有下游的很多分析没来得及做，也算是很大的遗憾了。其他做文献复现的两位同学都做的非常全面，工作量也很大。不过最起码每一步整得也算是明白，不至于抄了代码而一无所知了。收获还是不错的！希望大家天天开心哦

参考文献

[1] D. Li, X. Shu, P. Zhu, and D. Pei. Chromatin accessibility dynamics during cell fate reprogramming. *EMBO Reports*, 22(2):e51644, 2021. Q1.

[2] F. C. Grandi, H. Modi, L. Kampman, and M. R. Corces. Chromatin accessibility profiling by atac-seq. *Nature Protocols*, 17(6):1518–1552, 2022. Q1.