atac-seq pipeline

Amal Thomas

July 26, 2016

A computational pipeline for processing and analyzing atac-seq data

1 Overview

In our pipeline, we provide a handy work flow to process the atac-seq data and identify those regions with differential chromatin profile. The pipeline has mainly two parts:

- Pre-processing part: Here we process the input raw data and prepare it for the downstream analysis. The main steps include quality checks of the raw data, read alignment to the appropriate reference genome and subsequent filtering, peak calling and creating UCSC Track Hub.
- Post-processing part: Here we try to identify those chromosomal regions that show significant difference in the chromatin accessibility.

Users are assumed to have prior experience with UNIX/Linux environment and common bioinformatics tools.

2 Pre-processing

2.1 Mapping reads

Input files: The input to our pipeline is raw next generation sequencing data in the FASTQ format. The raw files from Sequence Read Archive (SRA), have to be decompressed and properly split to generate the right FASTQ files.

Quality check: `FastQC` provides an easy way to analyse the quality of raw sequencing data. This will give us a hint whether our data has any experimental error. One can employ the tool by the following command:

```
fastqc <read1.fastq> <read2.fastq> -o <outputdir>
```

Removing adapters: Sequencing adapters can be easily removed using `trimgalore` program. Typical command for paired end reads is:

```
trim_galore --paired   <read1.fastq>  <read2.fastq>  \
  --output_dir  <output directory>
```

Optional arguments:
To remove low quality reads use –quality parameter:

```
--quality <cutoff>
```

Mapping: The trimmed read are mapped to respective genome using `bowtie2`. Since the minimum distance between two Tn5 binding sites are about 38bp, only fragment size greater than this is kept.

```
bowtie2  -X 2000 --fr --no-discordant --no-mixed \
--minins 38  -x <bowtie-index> -1 <read1_trimmed.fastq> \
-2 <read2_trimmed.fastq> -S <read_mapped.sam>
```

Create sorted bam file: The output of mapping is in sam format, which is human readable. The downside of sam format is the huge file size. We can use samtools to convert samfile into smaller and compressed format called bam.

```
samtools view -bS read_mapped.sam \
 |samtools sort - read_mapped.sorted.bam
```

3 Qualtiy controls

Remove unwanted chromosome: Here we keep only those reads that is mapped to standard chromosomes. Reads mapped to mitochondrial DNA are also filtered out.

```
samtools view -h read_mapped.sorted.bam \
|awk ’substr($0,1,1) == "@"|| (length($3)<=5 && \
$3!="chrM" && $3 != "*"){{print $0}}’ \
|samtools view -bS - > read_mapped.sorted.chr.bam
```

Remove duplicate reads: PCR duplicated reads are removed using Picard Tools `MarkDuplicates`

```
java -Xmx2g -jar MarkDuplicates.jar \
INPUT = read_mapped.sorted.chr.bam \
METRICS_FILE=read_mapped.markdup.metrics" \
OUTPUT=read_mapped.sorted.chr.nodup.bam" \
REMOVE_DUPLICATES=true \
ASSUME_SORTED=true
```

Remove low quality mapped reads: Reads with Phred quality scores less than 30 are filtered out using the command:

```
samtools view -b -h -q 30 read_mapped.sorted.chr.nodup.bam \
> read_mapped.sorted.chr.nodup.filt.bam
```

The output bam file can be indexed using:

```
samtools index read_mapped.sorted.chr.nodup.filt.bam
```

4 Peak Calling

For each read to represent the center of Tn5 binding site we shift the + strands by +4 bp, and all reads aligning to the strand by 5 bp. Here for convinience we convert the bam file to bed format and do the shifting of cordinates.

```
bamToBed -i read_mapped.sorted.chr.nodup.filt.bam > temp.bed

samtools view  read_mapped.sorted.chr.nodup.filt.bam \
| awk ’BEGIN{OFS="\t"}{print $1,$9}’ > insert.temp
paste temp.bed insert.temp | \
awk ’BEGIN{OFS="\t"}{split($4,name,"/");if(name[1]== $7 && $6 == "+" )\
{print $1,$2+4,$3,$4,$8,$6}  \
else if(name[1]== $7 && $6 == "-" ){print $1,$2,$3-5,$4,$8,$6} \
> read_mapped.sorted.chr.nodup.filt.shift.bed

rm $INPUT_FILE".temp.bed" $INPUT_FILE".insert.temp"
```

Now we are ready to perform peak calling. For this we will use peak caller called MACS2. The shifted bed file is provided us the input for the program.

```
macs2 callpeak --gsize hs \
      -f BED \
         --treatment read_mapped.filt.sorted.chr.nodup.shift.bed  \
         --outdir peaks/ \
         --name read_mapped \
         --keep-dup all \
         --call-summits \
         --shift -100 --extsize 200 \
         --nomodel --nolambda \
         --verbose 3
```

The –shift -100 parameter will move the reads toward 3' to 5' direction and –extsize parameter will extend the 3' end to make 200 bp reads. This will ensure that center of each reads correspond to the original Tn5 cutting sites. For *e.g.*
Original reads:

chr1	500	550	read1	.	+
chr1	700	750	read2	.	-

Applying –shift -100:

chr1	400	450	read1	.	+
chr1	800	850	read2	.	-

Applying –extsize 200:

chr1	400	600	read1	.	+
chr1	650	850	read2	.	-

Remove blacklisted regions: We have to remove some problematic regions (microsatellites, centromeres, telomeres etc.) from the identified peaks. Encode blacklisted regions can be downloaded from this website.

5 Create track hub