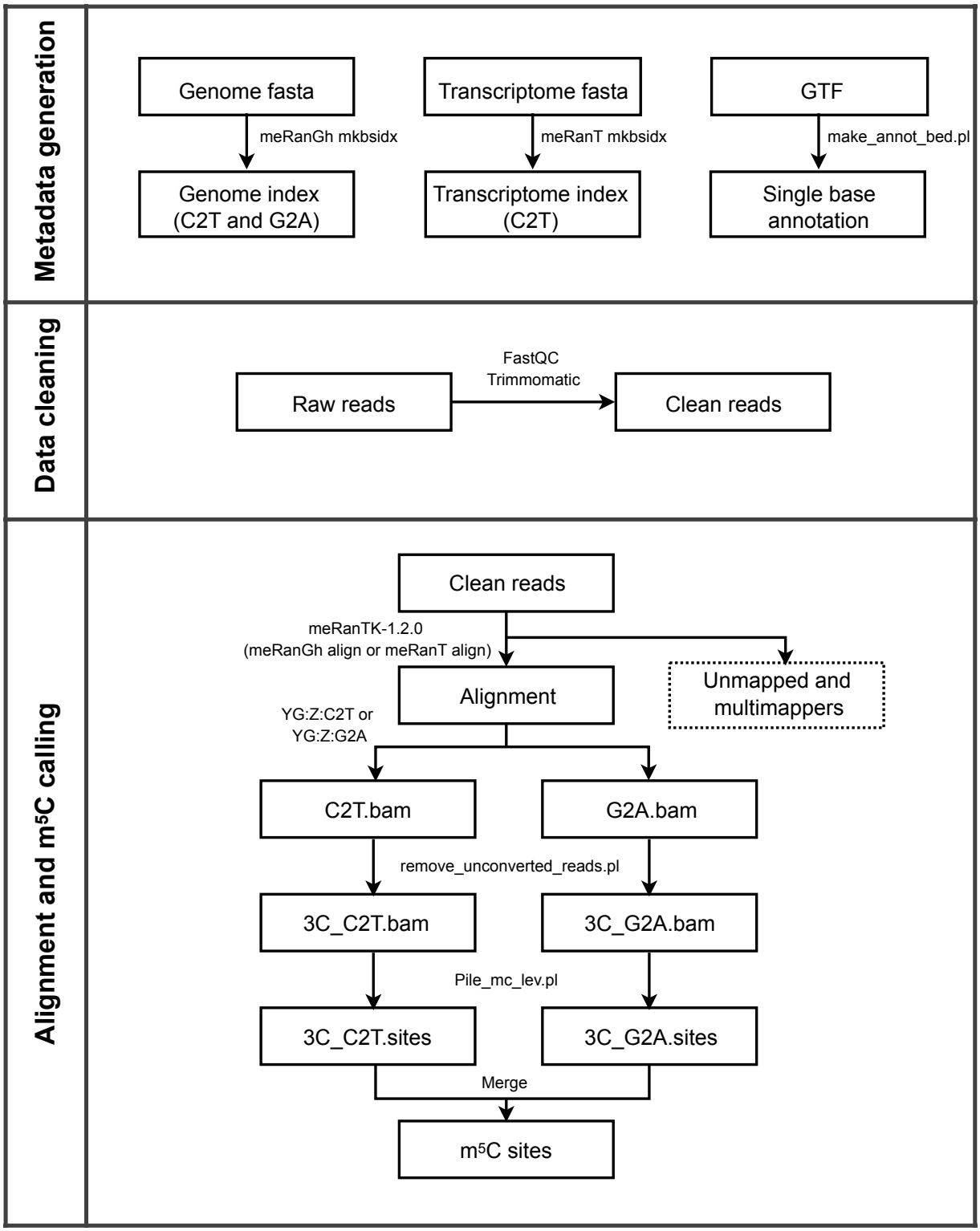
bsRNA-seq-m5C calling pipeline (v1)

Release date: October 2019, He-Na Zhang

1. Overview



2. Software

Quality control and formatting	FastQC v0.11.5 Trimmomatic v0.36 Cutadapt v1.18
Index and Mapping	meRanTK v1.2.0 (meRanGh and meRanT)
Bam processing	Samtools v1.6
R packages	clusterProfiler v3.10.1 ggseqlogo v0.1 Mfuzz v2.42.0 methylKit v1.8.1 RNAModR v0.1.1 ggplot2 v3.2.1

3. Customized scripts

Name	Usage
make_annot_bed.pl	Annotate each base in genePred
remove_unconverted_reads.pl	Remove reads with multiple unconverted Cs (default >3)
Pile_mc_lev.pl	Call m5C sites from bam files
SamByChr.sh	Call m5C sites from each chromosome
merge_sites.py	Merge m5C sites from multiple samples

4. Metadata

Name	Source	Comment
hg38_all.fa	UCSC	
ERCC92.fa	www-s.nist.gov	
R-luc.fa	(Bhattacharyya, Habermacher et al. 2006	
rDNA.fa	NCBI(U13369.1)	
hg38-tRNAs.bed	GtRNAdb	
hg38_gencode_V28_basic.genePred	UCSC	genePred annotation
hg38_gencode_V28_basic.annot.bed	make_annot_bed.pl	Genomic bases annotated by transcript

5. Pipeline

5.0 Metadata preparation

This step generates genome metadata index for RNA BS-seq mapping, pileup and site calling.

```
#Merge genome fasta with spike-in if needed

cat hg38_all.fa ERCC92.fa R-Luc.fa >Combined_hg38_ERCC_Rluc.fa

#C2T and G2A conversion of genome

meRanGh mkbsidx -t 4 -fa Combined_hg38_ERCC_Rluc.fa -id ./BSgenomeIDX
```

This step generates tRNA and rRNA metadata index for RNA BS-seq mapping, pileup and site calling.

```
# adding 100 nt 5' and 3' flanking regions.

modBed12.pl hg38-tRNAs.bed hg38-tRNAs.pre100.bed 100

#get fasta file from bed12, including intron

bedtools getfasta -name -s -fi hg38_all.fa -bed hg38-tRNAs.pre100.bed | \

perl -alne '@a=split(/:/,$F[0],2);print $a[0]' \

>hg38-tRNAs.pre100.fa

cat rRNA.fa hg38-tRNAs.pre100.fa > Pre-tRNA_and_rRNA.fa

#C2T and G2A conversion of genome

meRanT mkbsidx -fa Pre-tRNA_and_rRNA.fa -id \ ./00_Pre-tRNA_and_rRNA_BSgenomeIDX
```

5.1 Data cleaning

Raw reads were subjected to FastQC. Low quality bases and adaptor sequences were removed using Trimmomatic. (B1 sample)

```
fastqc -o 00_fastqc B1_R1.fastq.gz B1_R2.fastq.gz


java -jar trimmomatic-0.36.jar PE -threads 8 -phred33 B1_R1.fastq.gz B1_R2.fastq.gz \

B1_R1.paired.fq B1_R1.unpaired.fq B1_R2.paired.fq B1_R2.unpaired.fq \

ILLUMINACLIP:01_all_adapter.fa:2:30:10:8:true LEADING:3 TRAILING:3 \

SLIDINGWINDOW:4:20 CROP:97 HEADCROP:12  MINLEN:50
```

5.2 Map to Genome

Forward and reverse reads were C-to-T and G-to-A converted, respectively, and mapped to the appropriate converted reference using meRanGh in MeRanTK. Only uniquely mapped reads were retained and replaced by the original unconverted reads. (B1 sample)

```
meRanGh align -o B1 -f B1_R2.paired.fq -r B1_R1.paired.fq \

-t 16 -S B1.sam -un -ud ./B1 -MM -fmo \

-id BSgenomeIDX -GTF Combined_hg38_ERCC_Rluc.gtf -bg -mbp
```

5.3 Merge BAM files (optional)

If you have multiple BAM files, you can merge them before sites calling.

```
samtools merge B.bam B1/B1_sorted.bam […]

samtools sort B.bam B.sort.bam

samtools index B.sort.bam
```

5.4 Split to watson and crick BAM files

Split raw mapping BAM files to watson and crick BAM files using reads tag (YG:Z:C2T or YG:Z:G2A).

```
samtools view B.sort.bam | grep YG:Z:C2T > B_C2T.sam

samtools view B.sort.bam | grep YG:Z:G2A > B_G2A.sam

samtools view -H B.sort.bam > B.header

cat B.header B_C2T.sam > B.C2T.sam

cat B.header B_G2A.sam > B.G2A.sam

samtools view -bS B.C2T.sam > B.C2T.bam

samtools view -bS B.G2A.sam > B.G2A.bam

rm *sam

samtools index *.bam
```

5.5 Remove >3C reads

To remove background non-conversion, reads containing more than three unconverted cytosines were removed from the bam files.

```
perl remove_unconverted_reads.pl -i B.C2T.bam -t watson -n 3 | \

samtools view -bh - >B_3C.C2T.bam

perl remove_unconverted_reads.pl -i B.G2A.bam -t crick -n 3 | \

samtools view -bh - >B_3C.G2A.bam

samtools index *.bam
```

5.6 Call candidate sites

Call sites from the pileups with different strand (watson or crick) BAM files.
This script is used for a small number of sites in fixed areas.

```
perl Pile_mc_lev.pl -i B_3C.C2T.bam -t watson -r Combined_hg38_ERCC_Rluc.fa --rlength 101 \

--minBQ 30 --overhang 6 --depth 20000000 --reads 1 --cRatio 0 --variants 0 >B_3C.plus.sites

perl Pile_mc_lev.pl -i B_3C.G2A.bam -t crick-r Combined_hg38_ERCC_Rluc.fa --rlength 101 \

--minBQ 30 --overhang 6 --depth 20000000 --reads 1 --cRatio 0 --variants 0 >B_3C.minus.sites

cat B_3C.plus.sites B_3C.minus.sites >B_3C.sites
```

I highly recommend you use call sites on different chromosomes at the same time below.

```
./SamByChr.sh B_3C.C2T.bam SamByChr watson 4

./SamByChr.sh B_3C.G2A.bam SamByChr crick 4

cat SamByChr/* >B_3C.sites
```

You can use the following script to calculate signal ratio for each sites in each replicate (you need to recall sites before removing >3C reads using the above scripts).

```
# combie sites that are detected in at least one sample

python merge_sites.py BCE.sites 6 \

B_3C.sites C_3C.sites E_3C.sites B_raw.sites C_raw.sites E_raw.sites

# calculate signal ratio for each site in each replicate

awk -v OFS="\t" '{print $0,$2/$17,$7/$22,$12/$27}' BCE.sites >BCE.signal_ratio.sites
```

Then you can filter candidate sites as you need.