Detecting m^{5}C sites using BS-seq with IVT as negative control

Summary

m^{5}C is another extensively studied RNA modification though it is much rarer in transcriptome. We calibrated m^{5}C mapping and investigated the controversial results by integrating the IVT RNA library as a negative control during BS-seq. The data processes were followed the public pipeline "RNA-m5C", and then we integrated IVT sample as negative control. False positive rate and fisher's exact test were used to filter the m^{5}C sites. Only sites within exon regions were treated as high confidence m^{5}C sites.

Requirements

Unix/Linux based operating system (tested with CentOS release 6.10)
Perl (tested with version 5.26.2)
R (tested with version 3.5.2)
cutadapt (tested with version 1.15)
Trimmomatic (tested with version 0.36)
fastqc (tested with version 0.11.5)
hisat2 (tested with version 2.1.0)
samtools (tested with version 1.6)
RNA-m5C (https://github.com/SYSU-zhanglab/RNA-m5C)

Reference genome

GRCh37 (human)

Step-by-step instructions

The raw reads were removed adaptors and mapping to reference genome under the pipeline of RNA-m5C (https://github.com/SYSU-zhanglab/RNA-m5C). Same parameters were used in cellular mRNA and *in vitro* transcribed RNA (IVT RNA).

Steps 1-4 were under the guidance of RNA-m5C.

1. Remove adaptors and quality control

Paired-end raw reads were removed adaptors by cutadapt, the reads shorted than 15nt were discarded. fastqc was used to check the quality of clean reads.

cut adaptor under the guidance of RNA-m5C

```
## for mRNA
cutadapt -a AGATCGGAAGAGCACACGTCT -A AGATCGGAAGAGCGTCGTGT -O 3 -m 15 -e 0.25 -q 25 --trim-n -o mRNA_R1.cutout.fq -p mRNA_R2.cutout.fq rawdata_mRNA_R1.fq.gz rawdata_mRNA_R2.fq.gz
java -jar trimmomatic-0.36.jar PE -phred33 mRNA_R1.cutout.fq mRNA_R2.cutout.fq mRNA_rev.fastq mRNA_rev.UP.fq mRNA_fwd.fastq mRNA_fwd.UP.fq HEADCROP:10 SLIDINGWINDOW:4:22 AVGQUAL:25 MINLEN:40
## for IVT
cutadapt -a GATCGGAAGAGCACACGTCT -A AGATCGGAAGAGCGTCGTGT -O 3 -m 15 -e 0.25 -q 25 --trim-n -o IVT_R1.cutout.fq -p IVT_R2.cutout.fq rawdata_IVT_R1.fq.gz rawdata_IVT_R2.fq.gz
java -jar /Path_to_Trimmomatic/Trimmomatic-0.36/trimmomatic-0.36.jar PE -phred33 IVT_R1.cutout.fq IVT_R2.cutout.fq IVT_rev.fastq IVT_rev.UP.fq IVT_fwd.fastq IVT_fwd.UP.fq HEADCROP:10
```

SLIDINGWINDOW:4:22 AVGQUAL:25 MINLEN:40

2. Map to reference genome using RNA-m5C pipeline

Index construction under the guidance of RNA-m5C

for mRNA

python /path_to_RNA-m5C/2_m5C_step-by-step_hisat2/BS_hisat2.py -F mRNA_fwd.fastq -R mRNA_rev.fastq -o mRNA_hisat2 -I /Path_to_index/HISAT2_INDEX --index-prefix HISAT2 --hisat2-path /Path_to_hisat2/hisat2-2.1.0 --del-convert --del-sam 2>> mRNA_log

python /path_to_RNA-m5C/3_m5C_step-by-step_bowtie2/BS_bowtie2.py -F mRNA_fwd.unmapped.fastq -R mRNA_rev.unmapped.fastq -o mRNA_bowtie2 --bowtie2-path /Path_to_bowtie2/bowtie2-2.3.3/ -I Homo_sapiens.GRCh37.75.all.RNA.c2t -g Homo_sapiens.GRCh37.75.genelist --del-convert --del-sam --bowtie2-param parameter_file 2>> mRNA_bowtie2_log

python /path_to_RNA-m5C/3_m5C_step-by-step_bowtie2/Bam_transcriptome_to_genome_v1.0.py -i mRNA_bowtie2.bam -o mRNA_bowtie2_genome.bam -a /path_to_ref/Homo_sapiens.GRCh37.75.anno --dict Homo_sapiens.GRCh37.75.size 2>> mRNA_bowtie2_log

python /path_to_RNA-m5C/4_m5C_step-by-step_pileup/concat_bam.py -t 4 -i mRNA_hisat2.bam mRNA_bowtie2_genome.bam -o mRNA_merged.bam --sort --index 2>> mRNA_merge_log

for IVT

python /path_to_RNA-m5C/2_m5C_step-by-step_hisat2/BS_hisat2.py -F IVT_fwd.fastq -R IVT_rev.fastq -o IVT_hisat2 -I /Path_to_index/HISAT2_INDEX --index-prefix HISAT2 --hisat2-path /Path_to_hisat2/hisat2-2.1.0 --del-convert --del-sam 2>> IVT_log

python /path_to_RNA-m5C/3_m5C_step-by-step_bowtie2/BS_bowtie2.py -F IVT_fwd.unmapped.fastq -R IVT_rev.unmapped.fastq -o IVT_bowtie2 --bowtie2-path /Path_to_bowtie2/bowtie2-2.3.3/ -I Homo_sapiens.GRCh37.75.all.RNA.c2t -g Homo_sapiens.GRCh37.75.genelist --del-convert --del-sam --bowtie2-param parameter_file 2>> IVT_bowtie2_log

python /path_to_RNA-m5C/3_m5C_step-by-step_bowtie2/Bam_transcriptome_to_genome_v1.0.py -i IVT_bowtie2.bam -o IVT_bowtie2_genome.bam -a /path_to_ref /Homo_sapiens.GRCh37.75.anno --dict Homo_sapiens.GRCh37.75.size 2>> IVT_bowtie2_log

python /path_to_RNA-m5C/4_m5C_step-by-step_pileup/concat_bam.py -t 4 -i IVT_hisat2.bam IVT_bowtie2_genome.bam -o IVT_merged.bam --sort --index 2>> IVT_merge_log

3. Pileup using RNA-m5C pipeline

for mRNA

python /path_to_RNA-m5C/4_m5C_step-by-step_pileup/pileup_genome_multiprocessing_v1.4_pysam_v0.15.0.py -P 6 -f /path_to_ref/Homo_sapiens.GRCh37.75.dna_sm.primary_assembly.fa -i mRNA_merged.bam -o mRNA.pileups.tmp 2>> mRNA_pileup_log

python /path_to_RNA-m5C/4_m5C_step-by-step_pileup/m5C_pileup_formatter.py --db /path_to_ref/Homo_sapiens.GRCh37.75.noreundance.base -i mRNA.m5C.pileups.tmp -o mRNA.m5C.pileups.formatted.txt --CR mRNA_CR.txt 2>> mRNA_pileup_log

for IVT

python /path_to_RNA-m5C/4_m5C_step-by-step_pileup/pileup_genome_multiprocessing_v1.4_pysam_v0.15.0.py -P 6 -f /path_to_ref/Homo_sapiens.GRCh37.75.dna_sm.primary_assembly.fa -i IVT_merged.bam -o IVT.pileups.tmp 2>> IVT_pileup_log

python /path_to_RNA-m5C/4_m5C_step-by-step_pileup/m5C_pileup_formatter.py --db

/path_to_ref/Homo_sapiens.GRCh37.75.noreundance.base -i IVT.m5C.pileups.tmp -o IVT.m5C.pileups.formatted.txt --CR IVT_CR.txt 2>> IVT_pileup_log

4. Call m^{5}C sites using RNA-m5C pipeline

Call putative m^{5}C sites using two replicates of mRNA
python /RNA-m5C-master/5_m5C_step-by-step-call_site/m5C_caller_multiple.py -i sample.txt -o mRNA_2reps.csv -P 2 -c 10 -C 5 -r 0.05 -p 0.05 --method binomial
##cat sample.txt
##C5R2 mRNA.rep1.m5C.pileups.formatted.txt gene 20
##C5R3 mRNA.rep2.m5C.pileups.formatted.txt gene 20
python /RNA-m5C-master/5_m5C_step-by-step-call_site/m5C_intersection_single_r1.py -c 10 -C 5 -r 0.05 -p 0.05 -i mRNA_2reps.csv -o mRNA_2reps_intersect.csv

Call putative m^{5}C sites using two replicates of IVT
python /RNA-m5C-master/5_m5C_step-by-step-call_site/m5C_caller_multiple.py -i sample.txt -o IVT_2reps.csv -P 2 -c 10 -C 5 -r 0.05 -p 0.05 --method binomial
##cat sample.txt
##C5R2 IVT.rep1.m5C.pileups.formatted.txt gene 20
##C5R3 IVT.rep2.m5C.pileups.formatted.txt gene 20
python /RNA-m5C-master/5_m5C_step-by-step-call_site/m5C_intersection_single_r1.py -c 10 -C 5 -r 0.05 -p 0.05 -i IVT_2reps.csv -o IVT_2reps_intersect.csv

5. Call m^{5}C sites using IVT as negative control

Conduct following steps for two replicates
parameters:
#-n mRNA pileup file from step 3
#-i IVT pileup file from step 3
#-c C-content using RNA-m5C pipeline
#-d minimal depth
perl cal_fpr_m5c.pl -i IVT.m5C.pileups.formatted.txt -n mRNA.m5C.pileups.formatted.txt -c 20 -d 1 > mRNA_IVT.txt
awk ' $12!~/NA/&&$9>0{print}' mRNA_IVT .txt > mRNA_IVT_fail_convert .txt
Rscript fisher.r mRNA_IVT_fail_convert .txt > mRNA_IVT_fail_convert_fisher

6. Call high confidence m^{5}C sites between two replicates

##After conducting two replicates, we extracted the recurrent sites as high confidence m^{5}C sites.
parameters:
-r1 rep1 from step 5
-r2 rep2 from step 5
-c minimal number of unconverted C
-d minimal depth
-r minimal conversion failure rate
-f maximal false positive rate
-p maximal p-value in fisher exact test

```
perl call_m5c_highconf.pl -r1 mRNA_IVT_rep1_fail_convert_fisher -r2 mRNA_IVT_rep2_fail_convert_fisher -c 5 -d 10 -r 0.05 -f 0.2 -p 0.05 > m5c_highconf_sites
```