Pipeline for detecting m6A sites using m6A-ref-seq with FTO-treated sample as negative control

Summary
RNA ribonuclease MazF was reported to be sensitive to m6A modifications, which is able to cleave the ACA motif on the 5'-end but leave the (m6A)CA motif intact (Imanishi et al., 2017). Single-nucleotide resolution m6A detection method based on MazF, m6A-REF-seq and m6A-MAZTER-seq, were established by two teams independently (Zhang et al., 2019; Garcia-Campos et al., 2019). In m6A-RER-seq, FTO-treated samples were conducted as negative control to decrease the false positive sites. This file provided a pipeline for m6A identification using m6A-ref-seq and FTO-treated sample as negative control.

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)
HISAT2 (tested with version 2.1.0)
samtools (tested with version 1.6)

Reference genome:
GRCh37 (human): downloaded from http://ftp.ensembl.org/pub/release-75/

Step-by-step instructions

1. Remove adaptors using cutadapt

   # cut adaptor sequence for endogenous mRNA and FTO-treated samples. Reads less than 15nt were discarded.
   # example:
   cutadapt -a AGATCGGAAGAGCACACGTCTGAACTCCAGTCACNNNNNNATCTCGTATGCCGTCTTCTGCTTG -A GATCGTCGGACTGTAGAACTCTGAACGTGTAGATCTCGGTGGTCGCCGTATCATT -O 3 -m 15 -o mRNA_rep1_R1.cutout.fq.gz -p mRNA_rep1_R2.cutout.fq.gz mRNA_rep1_R1.fq.gz mRNA_rep1_R1.fq.gz

2. Prepare reference files

   2.0 Download the reference files

   Fasta and gtf files were downloaded and hisat2 index was generated using hisat2-build.

   2.1 Turn the assemble fasta file to two-line mode

   # This fasta file with two-line mode was used in downstream analysis
   perl prepare_ref_files/turn2line.pl Homo_sapiens.GRCh37.75.dna_sm.primary_assembly.fa > GRCh37.fa

   2.2 Scan and get location for each ACA motif in exon regions

   ## -g reference gtf file
   ## -f reference fasta file (two-line mode) from step 2.1
   perl prepare_ref_files/get_ref_motif.pl -g Homo_sapiens.GRCh37.75.gtf -f GRCh37.fa > GRCH37_motif_exon
   ## description of each column of GRCH37_motif_exon
   ## chromosome location strand

2.3 Get information of junction reads for each transcript

perl prepare_ref_files/get_junction_sites_trans.pl Homo_sapiens.GRCh37.75.gtf > GRCh37_junction_file

3. Map to reference genome

Map to reference genome and convert sam file to sorted bam file for each sample

Example:

```
hisat2 -p 8 -x /path_to_ref/hisat2_index/ref_hisat2 -1 mRNA_rep1_R1.cutout.fq.gz -2
mRNA_rep1_R2.cutout.fq.gz -S WT-mRNA_rep1.sam
samtools view -bS mRNA_rep1.sam > mRNA_rep1.bam
samtools sort -o -mRNA_rep1.sort.bam mRNA_rep1.bam
samtools view -b -F 256 -q 20 mRNA_rep1.sort.bam > mRNA_rep1.q20.mapped.sort.bam
samtools index mRNA_rep1.q20.mapped.sort.bam
```

4. Calculate the number of undigested reads for each ACA motif

Steps 4.0-4.1: Paired-end reads were combined to one fragment in bed format.

4.0 Convert bam file to temp bed file

Got the start and end of each fragment based on mapping positions of paired reads. The intron region was skipped based on CIGAR string of bam file. Each fragment was marked strand by flag information and only paired-end reads were kept.

samtools view file.q20.mapped.sort.bam | perl calculate_ACA/sam2bed_with_junction.pl > file.tmp

4.1 Convert tmp file to bed file

Generate bed file. Information of junction sites from gtf file was combined.

bed file was 0-based.

-g reference gtf file

-j junction sites file from step 2.3

-t tmp file from step 4.0

perl calculate_ACA/get_bed.pl -g gtf.file -j GRCh37_junction_file -t file.tmp > file.bed

description of each column of file.bed

chromosome start_location end_location strand reads_name

4.2 Locate and count ACA motif on the fragments

After digested by MazF, one RNA fragment was digested into two, generating 5' and 3' ends for each ACA motif. Based on the fragment sequence from above steps, we could get locations of ACA motifs on fragments. Internal ACA represents undigested site, while ACA on 5' terminal of fragment result in 5' end of digested site.

-f reference fasta file (two-line mode) from step 2.1

-m GRCH37_motif_exon from step 2.2

-b bed file from step 4.1

perl calculate_ACA/get_site_from_bed_with_ref.pl -f GRCh37.fa -m GRCH37_motif_exon -b file.bed > file.digest

4.3 Count digested/undigested numbers

Sometime, only one end (5' or 3' end) of a digested site was sequenced. Therefore, we also calculated the number of 3' end of each ACA. The maximum number of 5' and 3' ends was treated as digested number.
-m GRCH37_motif_exon from step 2.2
-t tmp file from step 4.0
-d digest file from step 4.2
perl calculate_ACA/cal_num_max.pl -t file.tmp -d file.digest -m GRCh37_motif_exon > file.num
description of each column of file.num
chromosome location strand number_of_undigested_reads number_of_digested_reads sum undigested_rate

5. Check SNP sites

5.1 Parsing pileup file

Mismatch sites should be considered when deciding m6A sites. If any base of ACA had been mutated, MazF could not digest this sequence.
Pileup step will take a long time, so it is better to do it just after bam file generated.
Mismatch information was parsed based on pileup file. Only sites with mismatch_rate > 0 were reported.
samtools mpileup --output-QNAME -f ref.fa file.q20.mapped.sort.bam -A -I -Q 0 | perl check_snp/reform_pileup.pl > file.mis.pileup
description of each column of file.mis.pileup
chromosome location ref_base number_of_sequenced_SE number_of_sequence_PE number_of_base_A number_of_base_T number_of_base_C number_of_base_G mismatch_rate

5.2 Remove mismatch sites

As mismatches of three bases of ACA would lead to false undigested signals, the mismatch number of three bases were counted.
-p pileup file from step 5.1
-n ACA num file from step 4.3
perl check_snp/rm_mismatch.pl -p file.mis.pileup -n file.num > file.mis.num
description of each column of file.mis.num
chromosome location strand number_of_undigested_reads number_of_digested_reads sum undigested_rate

Before conducting step 6, process steps1-5 for both mRNA and FTO-treated samples.

6. Calculate false positive rate (FPR) and conduct fisher exact test

6.1 Calculate false positive rate

The false-positive rate (FPR) is the ratio of false-positive results to total positive results, which was calculated through dividing undigestion rate of FTO-treated sample by that of the endogenous sample.
-i FTO num file after SNP checking
-n mRNA num file after SNP checking
perl FPR_test/cal_fpr.pl -i FTO.mis.num -n mRNA.mis.num > fpr_file
description of each column of fpr_file
chromosome location strand number_of_undigested _reads_in_FTO sum_in_FTO

undigested_rate_in_FTO number_of_undigested_reads_in_mRNA sum_in_mRNA
undigested_rate_in_mRNA fpr

6.2 Fisher's exact test

Rscript FPR_test/fisher_test.r fpr_file > fpr_fisher

description of each column of fpr_fisher

chromosome location strand number_of_undigested _reads_in_FTO sum_in_FTO
undigested_rate_in_FTO number_of_undigested_reads_in_mRNA sum_in_mRNA
undigested_rate_in_mRNA fpr fisher_p_value

Before conducting step7, process step 6 for three replicates.

7. Call high confidence m6A sites among three replicates

Call high confidence m6A sites from 3 replicates.

-r1 rep1_fpr_fisher

-r2 rep2_fpr_fisher

-r3 rep3_fpr_fisher

-d depth cutoff

-f fpr rate cutoff

-p p-value cutoff in fisher exact test

perl optional/call_m6A_highconf.pl -r1 rep1_fpr_fisher -r2 rep2_fpr_fisher -r3 rep3_fpr_fisher -d 10 -f 0.5 -p 0.05 > m6A_highconf.txt

description of each column of m6A_highconf.txt

chromosome location strand averaged_methylated_rate

8. (optional) Remove sites with continuous ACA motif in flanking sequences

Remove sites with continuous ACA motif in flanking 15nt, like 'ACACA' or 'ACANNACA', etc.

-f reference fasta file (two-line mode) from step 2.1

-s high confidence m6A sites from step 7

perl optinal/rm_multi_ACA.pl -f GRCh37.fa -s m6A_highconf.txt > m6A_highconf_rmACA.txt