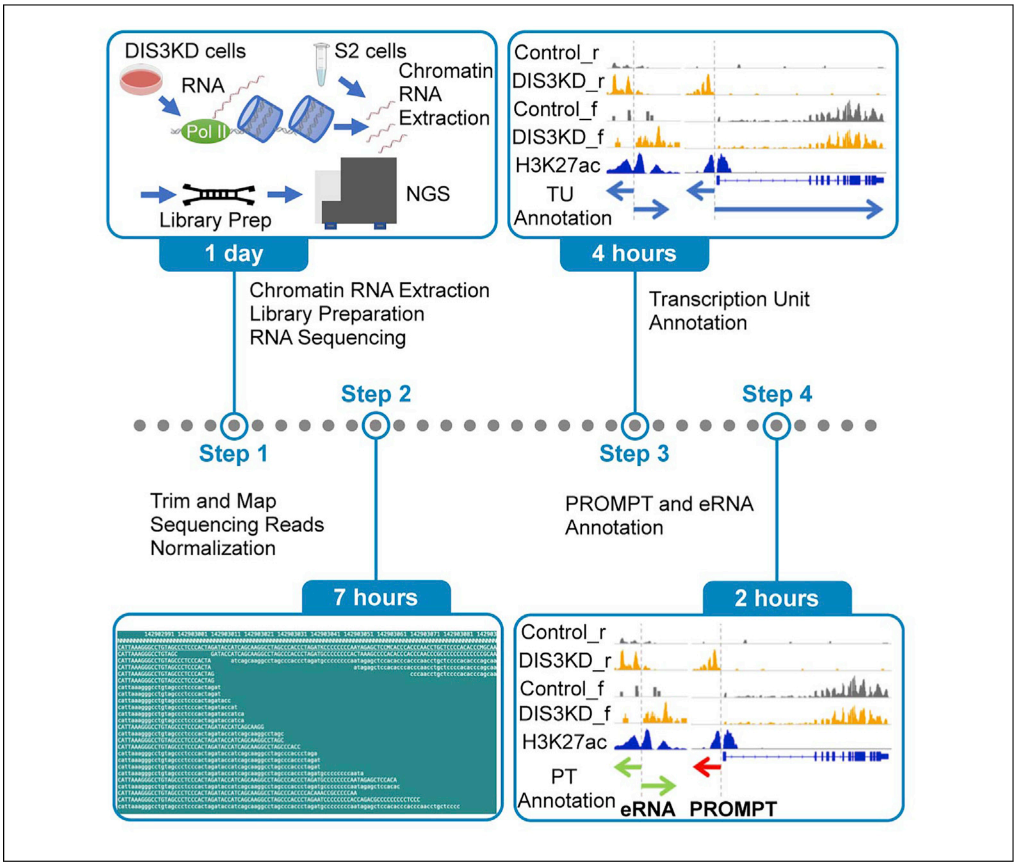
STAR Protocols

CellPress
OPEN ACCESS

Protocol

Defining pervasive transcription units using chromatin RNA-sequencing data



Pervasive transcripts (PTs) are difficult to detect by steady-state RNA-seq, because they are degraded immediately by the nuclear exosome complex. Here, we describe a protocol illustrating a bioinformatic pipeline for genome-wide PTs *de novo* annotation via chromatin-associated RNA-seq data upon DIS3 depletion. Compared to defining PTs by nascent RNA-seq such as TT-seq and PRO-seq, this protocol is more convenient and cost efficient. In addition, this protocol defines 3′-end of PTs more precisely, while reads from PRO-seq have a skew at the 5′-end.

Publisher's note: Undertaking any experimental protocol requires adherence to local institutional guidelines for laboratory safety and ethics.

Ziwei Guo, Xinhong Liu, Mo Chen

mochen@mail.tsinghua.edu.cn

Highlights

Efficient chromatin RNA extraction with spike-in RNA for RNA-seq normalization

Detection of accumulated PTs by chromatin RNA-seq upon Dis3 depletion

Annotate genome-wide PTs *de novo*

Bioinformatic pipeline for identification of sample-specific eRNAs and PROMPTs

Guo et al., STAR Protocols 3, 101442
June 17, 2022 © 2022 The Authors.
https://doi.org/10.1016/j.xpro.2022.101442

STAR Protocols

CellPress
OPEN ACCESS

Protocol

Defining pervasive transcription units using chromatin RNA-sequencing data

Ziwei Guo,[1,2] Xinhong Liu,[1] and Mo Chen[1,3,*]

[1]Tsinghua University School of Medicine, Beijing 100084, China

[2]Technical contact

[3]Lead contact

*Correspondence: guozw18@mails.tsinghua.edu.cn (Z.G.), mochen@mail.tsinghua.edu.cn (M.C.)

https://doi.org/10.1016/j.xpro.2022.101442

SUMMARY

Pervasive transcripts (PTs) are difficult to detect by steady-state RNA-seq, because they are degraded immediately by the nuclear exosome complex. Here, we describe a protocol illustrating a bioinformatic pipeline for genome-wide PTs *de novo* annotation via chromatin-associated RNA-seq data upon DIS3 depletion. Compared to defining PTs by nascent RNA-seq such as TT-seq and PRO-seq, this protocol is more convenient and cost efficient. In addition, this protocol defines 3′-end of PTs more precisely, while reads from PRO-seq have a skew at the 5′-end.

For complete details on the use and execution of this protocol, please refer to Liu et al. (2022).

BEFORE YOU BEGIN

Institutional permissions

All experiments involving animals and human samples must be approved by institutional permissions and national laws and regulations.

Preparing DIS3 deletion cells

Timing: ∼1–2 weeks

1. Use CRISPRi system (Stojic et al., 2018) specifically targeting *Dis3* for depletion.
   a. Use pLKO5.sgRNA.EFS.tRFP657 (Addgene #57824) vector to clone control and *Dis3*-specific sgRNAs.
   b. Transfect 70% confluent 293T cells with these lentiviral vector plasmids with both psPAX2 and pMD2.G packaging plasmids (Didier Trono, Addgene plasmids #12260 and #12259) using Neofect (Neofect # TF20121201).
   c. Collect culture supernatants containing lentiviruses 48 h and 72 h post transfection.
   d. Use these culture supernatants to infect sgRNAs to cell line expressing inducible dCas9-KRAB in a doxycycline dependent manner.
   e. Analyze the percentage of RFP positive cells using Accuri C6 Plus (BD Biosciences) to measure the infection efficiency.

Note: Two controls (sg4841 and sg9270) and two *Dis3* (sg*Dis3*-1 and sg*Dis3*-2) sgRNAs sequences are shown in the key resources table.

Alternatives: Other transfection reagents are acceptable.

2. Add 1 µg/mL doxycycline (dox) into culture medium to induce dCas9-KRAB expression for 72 h before harvesting cells.
3. Harvest cells and extract RNA. After reverse transcription, use *Actb* and *Dis3* primers to perform real-time qPCR to confirm the knockdown efficiency of *Dis3*.

   Alternatives: Harvest cells by SDS loading buffer and separate proteins by SDS-page electrophoresis. Transfer proteins to a nitrocellulose membrane and use antibody against DIS3 to test DIS3 protein level. β-ACTIN protein level can be used as the loading control.

Download software

4. Download and install Bowtie2 v.2.3.4.1 or above.

```
$wget     https://nchc.dl.sourceforge.net/project/bowtie-bio/bowtie2/2.3.4.1/bowtie2-
2.3.4.1-linux-x86_64.zip

$unzip bowtie2-2.3.5.1-linux-x86_64.zip
```

5. Download and install Trim_galore.

```
$wget https://github.com/FelixKrueger/TrimGalore/archive/0.6.4.tar.gz

$tar zxvf 0.6.4.tar.gz
```

6. Download and install STAR.

```
$ wget https://github.com/alexdobin/STAR/archive/2.7.1a.tar.gz

$ tar -xzf 2.7.1a.tar.gz

$ cd STAR-2.7.1a/source

$ make STAR
```

   Alternatives: One can use an open-source package and environment management system such as Anaconda (https://repo.anaconda.com/archive/Anaconda3-2021.11-Linux-x86_64.sh) to download above software. Other software to trim are available (e.g., Trimmomatic, http://www.usadellab.org/cms/index.php?page=trimmomatic).

KEY RESOURCES TABLE

REAGENT or RESOURCE	SOURCE	IDENTIFIER
Chemicals, peptides, and recombinant proteins		
Doxycycline	Solarbio	Cat#D8960-25g
Puromycin	Solarbio	Cat#P8230-25mg
G418	InvivoGen	Cat#ant-gn-1
TRIzol™ Reagent	Invitrogen™	Cat#15596018
Agencourt AMPure XP	Beckman Coulter	Cat#a63881
Critical commercial assays		
VAHTS Total RNA-seq (H/M/R) Library Prep Kit for Illumina	Vazyme	Cat#NR603
VAHTS RNA Adapters set3 for Illumina	Vazyme	Cat#N809
Deposited data		
RNA-seq data	Liu et al. (2022)	GEO: GSE162829
ChIP-seq data	Liu et al. (2022)	GEO: GSE162842

(*Continued on next page*)

Continued		
REAGENT or RESOURCE	SOURCE	IDENTIFIER
Experimental models: Cell lines		
HEK-293T	ATCC	CRL-11268
PDA cell	David et al. (2016)	N/A
Oligonucleotides		
sg9270 GCTTTCACGGAGGTTCGACG	Doench et al. (2016)	N/A
sg4841 ATGTTGCAGTTCGGCTCGAT	Doench et al. (2016)	N/A
sg*Dis3*-1 GGCCGTCAGCTAAGAAACCG	Horlbeck et al. (2016)	N/A
sg*Dis3*-2 GGCCTCGCGGCGTGGGATCC	Horlbeck et al. (2016)	N/A
Dis3-qPCR-F TTGATTCGGCGGCTATGAATG	Liu et al. (2022)	N/A
Dis3-qPCR-R TCAAAGCCCCCTTTTCAATCC	Liu et al. (2022)	N/A
Actb-qPCR-F CATTGCTGACAGGATGCAGAAGG	Liu et al. (2022)	N/A
Actb-qPCR-R TGCTGGAAGGTGGACAGTGAGG	Liu et al. (2022)	N/A
Gapdh-exon-F GTTGAGGTCAATGAAGGGGT	N/A	N/A
Gapdh-exon-R CCTCGTCCCGTAGACAAAATG	N/A	N/A
45s-F cctatctcgcttgtttctccc	N/A	N/A
45s-R gaaccactgagaaaagtgcg	N/A	N/A
Software and algorithms		
Bowtie2 version 2.3.4.1	Langmead and Salzberg (2012)	http://bowtie-bio.sourceforge.net/bowtie2/index.shtml
SAMTools version 1.9	Li et al. (2009)	http://samtools.sourceforge.net/
RStudio version 1.1.447	N/A	https://rstudio.com/
R version 3.5.0	N/A	https://www.r-project.org/
Trim_galore version 0.6.4	Babraham Institute	https://www.bioinformatics.babraham.ac.uk/projects/trim_galore/
STAR version 2.7.1a	Dobin and Gingeras (2015)	https://github.com/alexdobin/STAR/archive/2.7.1a.tar.gz $ tar -xzf 2.7.1a.tar.gz
FeatureCounts version 1.6.4	Liao et al. (2014)	https://sourceforge.net/projects/subread/files/subread-2.0.2/
Deeptools version 3.1.3	Ramirez et al. (2016)	https://deeptools.readthedocs.io/en/develop/
EdgeR	Robinson et al. (2010)	https://bioconductor.org/packages/release/bioc/html/edgeR.html
BEDTools version 2.29.0	Quinlan and Hall (2010)	https://github.com/arq5x/bedtools2/archive/v2.29.0.tar.gz
MACS2 version 2.1.1	Zhang et al., 2008	N/A
tiling-Array	Huber et al. (2006)	http://bioconductor.org/packages/release/bioc/html/tilingArray.html
GenoSTAN	Zacher et al. (2017)	N/A

MATERIALS AND EQUIPMENT

Lysis buffer		
Reagent	Final concentration	Amount
HEPES/KOH pH7.6 (1 M)	10 mM	500 μL
KCl (1 M)	15 mM	750 μL
EDTA (0.5 M)	2 mM	200 μL
Spermine (1 M)	0.15 mM	7.5 μL
Spermidine (1 M)	0.5 mM	25 μL
Glycerol (100%)	10%	5 mL

(*Continued on next page*)

Continued

Reagent	Final concentration	Amount
Sucrose (1 M)	0.3 M	15.015 mL
NP-40 (20%)	0.5%	1.25 mL
RNase-free H2O	n/a	28.5025 mL
Total	**n/a**	**50 mL**

Store at 4°C and keep in dark for up to 1 year. Add RNasin to 12.5 U/mL final concentration when necessary. Add spermine and spermidine when it will be used.

Cushion buffer

Reagent	Final concentration	Amount
HEPES/KOH pH7.6 (1 M)	10 mM	500 μL
KCl (1 M)	15 mM	750 μL
EDTA (0.5 M)	2 mM	200 μL
Spermine (1 M)	0.15 mM	7.5 μL
Spermidine (1 M)	0.5 mM	25 μL
Glycerol (100%)	10%	5 mL
Sucrose (2 M)	0.87 M	21.75 mL
RNase-free H2O	n/a	21.7675 mL
Total	**n/a**	**50 mL**

Store at 4°C and keep in dark for up to 1 year. Add RNasin to 12.5 U/mL final concentration when necessary. Add spermine and spermidine when it will be used.

Nuclear storage buffer

Reagent	Final concentration	Amount
HEPES/KOH pH7.6 (1 M)	10 mM	500 μL
KCl (1 M)	100 mM	5 mL
EDTA (0.5 M)	0.1 mM	10 μL
Spermine (1 M)	0.15 mM	7.5 μL
Spermidine (1 M)	0.5 mM	25 μL
Glycerol (100%)	10%	5 mL
RNase-free H2O	n/a	39.4575 mL
Total	**n/a**	**50 mL**

Store at 4°C and keep in dark for up to 1 year. Add RNasin to 12.5 U/mL final concentration when necessary. Add spermine and spermidine when it will be used.

2× NUN buffer

Reagent	Final concentration	Amount
HEPES/KOH pH7.6 (1 M)	50 mM	2.5 mL
NaCl (5 M)	0.6 M	6 mL
NP-40 (20%)	2%	5 mL
Urea (8 M)	2 M	12.5 mL
RNase-free H2O	n/a	24 mL
Total	**n/a**	**50 mL**

Store at 4°C and keep in dark for up to 1 year. Add RNasin to 12.5 U/mL final concentration when necessary. Add Urea when it will be used.

⚠ CRITICAL: KOH can cause chemical burns. Handle with care and avoid any contact with skin and eyes.

STEP-BY-STEP METHOD DETAILS

Chromatin RNA extraction with spike-in control

⏲ Timing: ~3–4 h

DIS3KD chromatin RNA are extracted using S2 cells as spike-in control.

1. Digest and wash cells with ice-cold PBS.
2. After centrifugation for 3 min at 300 g with a fixed angle rotor at 4°C, lyse cell pellets with lysis buffer and incubated for 10 min on ice.
3. Add 30% cushion buffer to the lysates in centrifugation tubes and spun nuclei for 15 min at 3 500 g with a fixed angle rotor at 4°C.
4. Wash the isolated nuclei with ice-cold PBS with 1 mM EDTA once and resuspend the nuclei in nuclear storage buffer.
5. Add one volume of 2× NUN buffer to one volume of nuclei in nuclear storage buffer for 30 min on ice.
6. After centrifugation for 30 min at 21 000 g with a fixed angle rotor at 4°C, wash chromatin pellets twice with 1× NUN buffer.
7. Resuspended the pellets in TRIzol for RNA extraction.
8. Add 0.08% *Drosophila* S2 cells as spike-in control.
9. Immediately extract RNA, or samples can be stored at −20°C for up to a year.
10. Extract high-quality RNA according to TRIzol manual. Troubleshooting 1 and 2.
11. Use Agilent 2100 Bioanalyzer and RT-qPCR to perform quality control (Figures 1A and 1B).

Alternatives: Cells from other species can also be used for normalization. We recommend using higher percentage of spike-in controls, such as 10%–20%.

⏸ **Pause point:** RNA samples can be store at −80°C for up to a year.

Note: Chromatin RNA samples do not contain 28s and 18s rRNA bands like RNA extracted from whole cells so their RNA integrity number (RIN) is low (Figure 1A) and they have higher level of 45s rRNA and lower level of mature mRNA (Figure 1B).

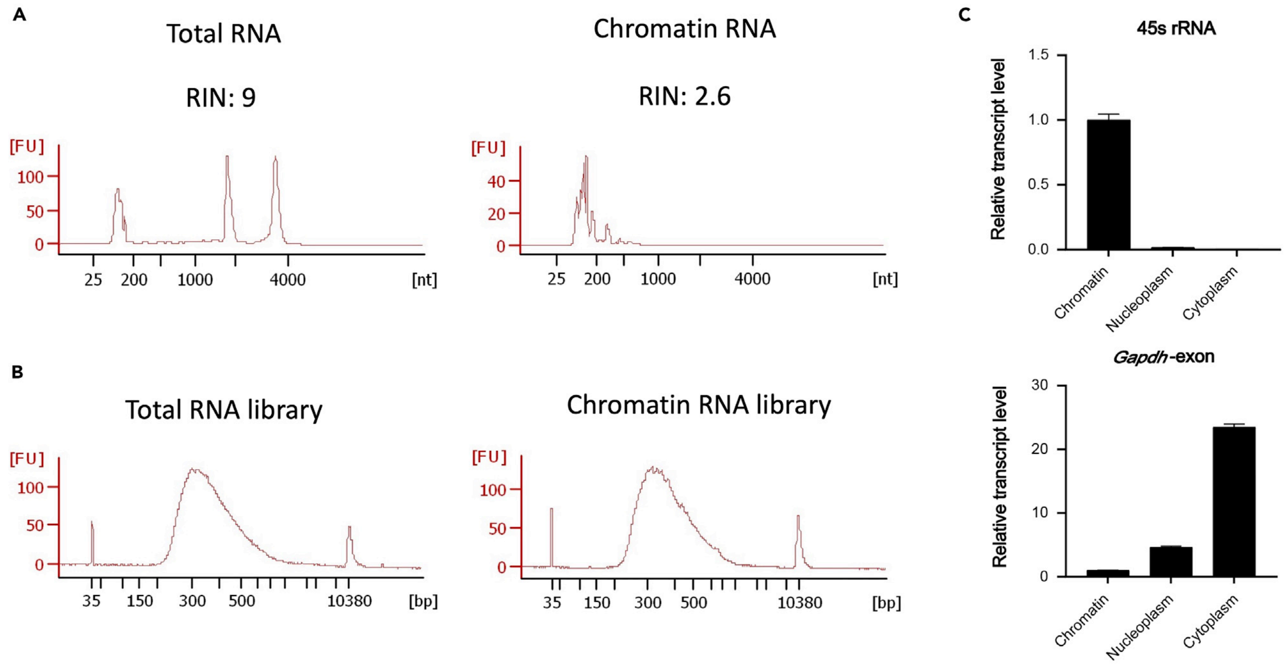


Figure 1. Quality control of chromatin RNA

(A) Characterization of RNA quality using the Agilent 2100 Bioanalyzer. The left profile has three peaks which stand for 5.8S and 5S (below 200 nt), 18s (around 2000 nt) and 28s (around 4000 nt) rRNAs. The right profile only has one major RNA peak at around 200 nt. The RINs are shown above.

(B) Relative abundance of the indicated RNAs in the three cellular compartments, determined by real-time qPCR.

(C) Bioanalyzer electropherograms of the sequencing libraries. The libraries are size-selected with AMPure XP beads. Error bars represent standard deviation.

Library preparation and RNA sequencing

Timing: ~8–10 h

Prepare library for chromatin RNA and do RNA sequencing.

12. 500 ng RNA of each sample should be used to prepare libraries.
13. Deplete rRNAs and prepare total RNA-seq libraries using the VAHTS Total RNA-seq (H/M/R) Library Prep Kit for Illumina (Vazyme, NR603-01) (Figure 1C).
14. Perform 150 bp paired-end high throughput RNA sequencing on Illumina HiSeq Xten platform.

Note: *Dis3* depletion can greatly increase the RNA level of PTs which are normally at much lower abundance than mRNAs. Therefore, only around 10–20 million reads per library is sufficient.

Note: In 1–14 steps, different samples should be processed at the same time by the same reagents and equipment to reduce batch effects.

Acquire H3K27ac ChIP-seq data

Timing: ~2 h

Download cell-line or tissue specific H3K27ac ChIP-seq data to identify active enhancer locations.

15. Download the raw sequencing data (https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE162842) in fq.gz format of H3K27ac ChIP-seq data generated from a primary pancreatic cancer cell line (Liu et al., 2022).

Alternatives: Download public tissue or cell line specific H3K27ac ChIP-seq data that fit the chromatin RNA-seq data.

Trim and map sequencing reads

Timing: ~4–6 h

Analyses of RNA-seq and ChIP-seq data.

The minimum computational requirement: this protocol requires at least 30 GB of RAM.

16. Download the mouse genome information and the gene annotation GTF file from GENCODE at https://www.gencodegenes.org/mouse/.
    a. Download the fasta files from "Genome sequence (GRCm39)" in the "Fasta files" table named "GRCm39.genome.fa.gz".
    b. Download the GTF file from "Comprehensive gene annotation" in the "GTF/GFF3 files" table named "gencode.vM28.chr_patch_hapl_scaff.annotation.gtf.gz".
17. Download the *Drosophila* genome information and the gene annotation GTF file from Ensembl at https://asia.ensembl.org/.
    a. Choose the species as *Drosophila melanogaster* and navigate to the "fasta" folder. Navigate to the "dna" folder and download the genome file named "Drosophila_melanogaster.BDGP6.32.dna.toplevel.fa.gz".
    b. Navigate to the "gtf" folder and download the gene annotation file named "Drosophila_melanogaster.BDGP6.32.105.gtf.gz".
18. Unzip genome and annotation files.

```
$gunzip FILEPATH/GRCm39.genome.fa.gz

$gunzip FILEPATH/gencode.vM28.chr_patch_hapl_scaff.annotation.gtf.gz

$gunzip FILEPATH/Drosophila_melanogaster.BDGP6.32.dna.toplevel.fa.gz

$gunzip FILEPATH/Drosophila_melanogaster.BDGP6.32.105.gtf.gz
```

19. Trim adapter and low-quality reads.

```
$trim_galore -q 25 –phred33 –length 50 -e 0.1 –fastqc –fastqc_args ‘‘-outdir./’’ –stringency 5
–paired –output_dir./

RAWDATA_DIRECTION/sg9270_R1.fq.gz RAWDATA_DIRECTION/sg9270_R2.fq.gz

$trim_galore -q 25 –phred33 –length 50 -e 0.1 –fastqc –fastqc_args ‘‘-outdir./’’ –stringency 5
–paired –output_dir./

RAWDATA_DIRECTION/sg4841_R1.fq.gz RAWDATA_DIRECTION/sg4841_R2.fq.gz

$trim_galore -q 25 –phred33 –length 50 -e 0.1 –fastqc –fastqc_args ‘‘-outdir./’’ –stringency 5
–paired –output_dir./

RAWDATA_DIRECTION/sgDis3_1_R1.fq.gz RAWDATA_DIRECTION/sgDis3_1_R2.fq.gz

$trim_galore -q 25 –phred33 –length 50 -e 0.1 –fastqc –fastqc_args ‘‘-outdir./’’ –stringency 5
–paired –output_dir./

RAWDATA_DIRECTION/sgDis3_2_R1.fq.gz RAWDATA_DIRECTION/sgDis3_2_R2.fq.gz

$trim_galore –fastqc –quality 20 –paired –phred33 –output_dir./

FILEPATH/H3K27ac_R1.fq.gz FILEPATH/H3K27ac_R2.fq.gz

$trim_galore –fastqc –quality 20 –paired –phred33 –output_dir./

FILEPATH/Input_R1.fq.gz FILEPATH/Input_R2.fq.gz
```

Note: Download the original files at https://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=GSE162829.

20. Build the mouse and *Drosophila* genome indexes for alignment.

```
$STAR –runThreadN 16 –runMode genomeGenerate –genomeDir./–

genomeFastaFiles FILEPATH/GRCm39.genome.fa –sjdbGTFfile

FILEPATH/gencode.vM28.chr_patch_hapl_scaff.annotation.gtf –sjdbOverhang 149

$STAR –runThreadN 16 –runMode genomeGenerate –genomeDir./–

genomeFastaFiles

FILEPATH/Drosophila_melanogaster.BDGP6.32.dna.toplevel.fa –

sjdbGTFfile FILEPATH/Drosophila_melanogaster.BDGP6.32.105.gtf –

sjdbOverhang 149

$bowtie2-build FILEPATH/GRCm39.genome.fa mm10
```

Note: "–sjdbOverhang" value should set as ReadLength – 1.

21. Align the chromatin RNA-seq and H3K27ac ChIP-seq data. Troubleshooting 3.
    a. Align the chromatin RNA-seq and H3K27ac ChIP-seq data to the mouse genome.

```
$for file in sg9270 sg4841 sgDis3_1 sgDis3_2
   do
   STAR –runThreadN 16 –outFilterMultimapScoreRange 1 –
   outFilterMultimapNmax 20 –outFilterMismatchNmax 10
   –alignIntronMax 500000 –alignMatesGapMax 1000000 –sjdbScore 2 -
   –alignSJDBoverhangMin 1 –genomeLoad NoSharedMemory –
   limitBAMsortRAM 20000000000 –c –outFilterMatchNminOverLread
   0.33 –outFilterScoreMinOverLread 0.33 –sjdbOverhang 149 –
   outSAMstrandField intronMotif –outSAMattributes NH HI NM MD AS
   XS –limitSjdbInsertNsj 2000000 –outSAMunmapped None –
   outSAMtype BAM SortedByCoordinate –outSAMheaderHD @HD VN:1.5 –
   twopassMode Basic –outSAMmultNmax 1 –genomeDir
   FILEPATH/STAR_INDEX_mm10 –readFilesCommand zcat –readFilesIn
   ${file}_R1_val_1.fq.gz ${file}_R2_val_2.fq.gz –sjdbGTFfile
   FILEPATH/gencode.vM28.chr_patch_hapl_scaff.annotation.gtf –
   outFileNamePrefix ./${file}_mm10_
   done
$bowtie2 -p 8 -x BOWTIE2_INDEX -1 H3K27ac_R1_val_1.fq.gz -2
H3K27ac_R2_val_2.fq.gz -S H3K27ac.sam –very-sensitive -X 2000
$bowtie2 -p 8 -x BOWTIE2_INDEX -1 Input_R1_val_1.fq.gz -2
Input_R2_val_2.fq.gz -S Input.sam –very-sensitive -X 2000
```

b. Align the chromatin RNA-seq to the *Drosophila* genome.

```
$for file in sg9270 sg4841 sgDis3_1 sgDis3_2
   do
   STAR –runThreadN 16 –outFilterMultimapScoreRange 1 –
   outFilterMultimapNmax 20 –outFilterMismatchNmax 10 –
   alignIntronMax 500000 –alignMatesGapMax 1000000 –sjdbScore 2 –
   alignSJDBoverhangMin 1 –genomeLoad NoSharedMemory –
   limitBAMsortRAM 20000000000 –c –outFilterMatchNminOverLread
   0.33 –outFilterScoreMinOverLread 0.33 –sjdbOverhang 149 –
   outSAMstrandField intronMotif –outSAMattributes NH HI NM MD AS
   XS –limitSjdbInsertNsj 2000000 –outSAMunmapped None –
   outSAMtype BAM SortedByCoordinate –outSAMheaderHD @HD VN:1.5 –
   twopassMode Basic –outSAMmultNmax 1 –genomeDir
   FILEPATH/STAR_INDEX_dm6 –readFilesCommand zcat –readFilesIn
   ${file}_R1_val_1.fq.gz ${file}_R2_val_2.fq.gz –sjdbGTFfile
```

```
    FILEPATH/Drosophila_melanogaster.BDGP6.32.105.gtf –
    outFileNamePrefix ./${file}_dm6_
    done
```

c. Remove unmapped or unpaired reads and generate indexes of filtered bam files using Samtools.

```
$for file in sg9270 sg4841 sgDis3_1 sgDis3_2
    do
    samtools view -b -h -F 4 -f 3
    ${file}_mm10_Aligned.sortedByCoord.out.bam -o
    ${file}_mm10_filter.bam
    samtools view -b -h -F 4 -f 3
    ${file}_dm6_Aligned.sortedByCoord.out.bam -o
    ${file}_dm6_filter.bam
    samtools index ${file}_mm10_filter.bam
    samtools index ${file}_dm6_filter.bam
    done
$for file in H3K27ac Input
    do
    samtools view -b -h -F 1028 -f 3 ${flie}.sam -o
    ${flie}_filter.bam
    samtools sort ${flie}_filter.bam -@ 16 >
    ${flie}_filter_sorted.bam
    samtools index ${flie}_filter_sorted.bam
    done
```

Note: “–genomeDir" input the path of STAR index generated in step 20.

Note: BOWTIE2_INDEX is generated in step 20.

Note: The input sam/bam files in step c is the output sam/bam files in step a.

Calculate SizeFactor of each sample using spike-in control

⏱ Timing: ~1 h

Use spike-in control to normalize samples.

22. Calculate *Drosophila* gene counts.

```
$featureCounts -p -g gene_name -a FILEPATH/Drosophila_melanogaster.BDGP6.32.105.gtf -s 2 -o
dm6_Dis3_RNAseqcount.txt sg9270_dm6_filter.bam sg4841_dm6_filter.bam sgDis3_1_dm6_filter.-
bam sgDis3_2_dm6_filter.bam
```

Note: Input trimmed bam file of each sample separated by space.

Note: Uses strand-specific RNA-seq Library Prep Kit. Use "-s 2" to calculate strand-specific gene counts.

23. Calculate sizeFactor of each sample. The sizeFactors of sg4841, sg9270, sgDis3_1 and sgDis3_2 samples are 0.9232670, 1.1442665, 0.9295284 and 1.0183166.

```
>library(edgeR)

>gene <- read.table(file = "dm6_Dis3_RNAseqcount.txt", sep = "\t", header = T, skip =1)

>Dis3_gene = cbind(gene[7:10])

>Geneid <- rbind(gene[1])

>rownames(Dis3) <- Geneid$Geneid

>countMatrix <- as.matrix(Dis3)

>head(countMatrix)

>group <- factor(c("c","c","t","t"))

>y <- DGEList(counts=countMatrix,group=group)

>y

#####FILTER#####

>keep <- rowSums(cpm(y)>0) >= 2;

>y <- y[keep,keep.lib.sizes=FALSE];

>dim(y)

#####sizeFactors#####

>calcNormFactors(y)

#sg4841 = 0.9232670

#sg9270 = 1.1442665

#sgDis3_1 = 0.9295284

#sgDis3_2 = 1.0183166
```

Transcription unit (TU) annotation

Timing: ~4 h

Use Poisson Log-normal emission distribution to identify transcription units.

24. Calculate the genome-wide coverage from each chromatin RNA-seq sample in consecutive 200 bp bins throughout the genome.
    a. Generate strand-specific BW files of chromatin RNA-seq samples.

```
$for file in sg9270 sg4841 sgDis3_1 sgDis3_2

   do

   bamCoverage -binSize 10 -normalizeUsing RPKM -bam ${file}_filter.bam -filterRNAstrand forward -o ${file}_filter_fwd.bw

   bamCoverage -binSize 10 -normalizeUsing RPKM -bam ${file}_filter.bam -filterRNAstrand reverse -o ${file}_filter_re.bw

   done
```

b. Use the mean value of the two samples from *Dis3* depletion to generate bedgraph files representing (+) and (-) strand.

```
$bigwigCompare -b1 Dis3_1_filter_fwd.bw -b2 Dis3_2_filter_fwd.bw -skipNAs -operation mean -bs 200 -of bedgraph -o Dis3_fwd.bedgraph

$bigwigCompare -b1 Dis3_1_filter_re.bw -b2 Dis3_2_filter_re.bw -skipNAs -operation mean -bs 200 -of bedgraph -o Dis3_re.bedgraph
```

Note: Other normalization methods normalize for sequencing depth are acceptable. The aim is to define transcription unit using *Dis3* depletion samples, so do not need to use spike-in control to normalize.

Note: Use "–filterRNAstrand" to generate strand-specific BW files.

25. Use Poisson Log-normal emission distribution to divide bins from step 24 to "transcribed" state and "untranscribed" state.

```
>if (!requireNamespace(’’BiocManager’’, quietly = TRUE))

> install.packages("BiocManager")

>BiocManager::install("STAN")

>library(STAN)

>bedgraph <- read.table("Dis3_fwd.bedgraph",header = F)

>sampleNames <- c("Chr","start","end","RPKM")

>names(bedgraph) <- sampleNames

>head(bedgraph)

>tmp <- list()

>RPKM <- matrix(bedgraph$RPKM)

>tmp$RPKM <- RPKM

>hmm_nb <- initHMM(tmp, 2, "PoissonLogNormal")

>hmm_fitted_nb <- fitHMM(tmp, hmm_nb, maxIters=10)

>viterbi_nb <- getViterbi(hmm_fitted_nb, tmp)

>RPKM_state <- cbind(as.vector(unlist(viterbi_nb[1])),as.vector(unlist(tmp[1])))

>bedgraph_state <- cbind(bedgraph,RPKM_state)

>bedgraph_state1 <- bedgraph_state[which(bedgraph_state$‘1‘=="1"),]

#transcribed state
```

```
>bedgraph_state2 <- bedgraph_state[which(bedgraph_state$‘1‘=="2"),]
#untranscribed state
>write.table(bedgraph_state1,file <- "Dis3_fwd_state1.bed",quote = F,
sep = "\t",row.names = F)
```

Note: Use forward strand as an example.

26. Merge contiguous transcribed bins within 200 bp as one TU.

```
$bedtools merge -d 200 -i Dis3_fwd_state1.bed |awk ’{OFS="\t"}{print
$1,$2,$3,".",".","+"}’ > Dis3_fwd_state1_merge.bed
$bedtools merge -d 200 -i Dis3_re_state1.bed |awk ’{OFS=’’\t’’}{print
$1,$2,$3,".",".","-"}’ > Dis3_re_state1_merge.bed
```

Select intergenic TUs overlapping with H3K27ac

Timing: ~4–6 h

Intergenic TUs overlapping with H3K27ac are defined as candidates of PTs.

27. Select intergenic TUs.
    a. Convert the mouse gene annotation file from GTF format to BED format.

```
$awk  -F  "\t"  ’{OFS="\t"}{match($9,"gene_name  \"(.+)\";  transcript_type",a)}{if
($3~/gene/)print($1,$4,$5,a[1],0,$7)}’    FILEPATH/gencode.vM28.chr_patch_hapl_scaff.
annotation.gtf > gencode.vM28.chr_patch_hapl_scaff.annotation.bed
```

    b. Remove TUs that overlap with coding genes on the same strand. Troubleshooting 4.

```
$bedtools intersect -s -v -a Dis3_fwd_state1_merge.bed -b
FILEPATH/gencode.vM21.chr.annotation.bed |uniq >
Dis3_fwd_intergenic.bed
$bedtools intersect -s -v -a Dis3_re_state1_merge.bed -b
FILEPATH/gencode.vM21.chr.annotation.bed |uniq >
Dis3_re_intergenic.bed
$cat Dis3_re_intergenic.bed Dis3_fwd_intergenic.bed >
Dis3_intergenic.bed
```

28. Find and select intergenic TUs overlapping with H3K27ac peak regions.
    a. Peak calling of H3K27ac ChIP-seq data.

```
$macs2 callpeak -B -t FILEPATH/H3K27ac_filter_sorted.bam -c
FILEPATH/Input_filter_sorted.bam -n H3K27ac –verbose 3 -g mm -B -q
1e-3 -f BEDPE –broad
```

b. Filter out TUs that do not overlap with H3K27ac peaks. Troubleshooting 4.

```
$bedtools intersect -wa -a Dis3_re_intergenic.bed -b
H3K27ac.broadPeak|uniq > Dis3_re_intergenic_H3K27ac.bed
$bedtools intersect -wa -a Dis3_fwd_intergenic.bed -b
H3K27ac.broadPeak|uniq > Dis3_fwd_intergenic_H3K27ac.bed
```

29. TUs start and end sites are refined to nucleotide precision.
    a. Define 400 bp bins located around start and end sites of the initially assigned TUs.

```
$awk '{OFS="\t"}{print($1,$2-200,$2+200)}'
Dis3_fwd_intergenic_H3K27ac.bed > Dis3_fwd_intergenic_H3K27ac_5end.bed
$awk '{OFS="\t"}{print($1,$3-200,$3+200)}'
Dis3_fwd_intergenic_H3K27ac.bed > Dis3_fwd_intergenic_H3K27ac_3end.bed
$awk '{OFS="\t"}{print($1,$2-200,$2+200)}'
Dis3_re_intergenic_H3K27ac.bed > Dis3_re_intergenic_H3K27ac_3end.bed
$awk '{OFS="\t"}{print($1,$3-200,$3+200)}'
Dis3_re_intergenic_H3K27ac.bed > Dis3_re_intergenic_H3K27ac_5end.bed
$samtools depth -a -b Dis3_fwd_intergenic_H3K27ac_5end.bed
sgDis3_1_mm10_filter.bam >
Dis3_1_fwd_intergenic_H3K27ac_5end_depth.txt
$samtools depth -a -b Dis3_fwd_intergenic_H3K27ac_3end.bed
sgDis3_1_mm10_filter.bam >
Dis3_1_fwd_intergenic_H3K27ac_3end_depth.txt
$samtools depth -a -b Dis3_re_intergenic_H3K27ac_5end.bed
sgDis3_1_mm10_filter.bam >
Dis3_1_reintergenic_H3K27ac_5end_depth.txt
$samtools depth -a -b Dis3_re_intergenic_H3K27ac_3end.bed
sgDis3_1_mm10_filter.bam >
Dis3_1_re_intergenic_H3K27ac_3end_depth.txt
$samtools depth -a -b Dis3_fwd_intergenic_H3K27ac_5end.bed
sgDis3_2_mm10_filter.bam >
Dis3_2_fwd_intergenic_H3K27ac_5end_depth.txt
$samtools depth -a -b Dis3_fwd_intergenic_H3K27ac_3end.bed
sgDis3_2_mm10_filter.bam >
Dis3_2_fwd_intergenic_H3K27ac_3end_depth.txt
$samtools depth -a -b Dis3_re_intergenic_H3K27ac_5end.bed
```

```
sgDis3_2_mm10_filter.bam >

Dis3_2_reintergenic_H3K27ac_5end_depth.txt

$samtools depth -a -b Dis3_re_intergenic_H3K27ac_3end.bed

sgDis3_2_mm10_filter.bam >

Dis3_2_re_intergenic_H3K27ac_3end_depth.txt
```

b. Find the locations of abrupt coverage increase or decrease within the 400 bp bins.

```
>library("tilingArray")

>library("davidTiling")

#####sizeFactor#####

>Dis3_1_sizeFactor <- 0.9295284

>Dis3_2_ sizeFactor <- 1.0183166

>rep_num <- 2

>rep1_path <- "Dis3_1_fwd_intergenic_H3K27ac_5end_depth.txt"

>rep2_path <- "Dis3_2_fwd_intergenic_H3K27ac_5end_depth.txt"

>output_path <- "./Dis3_fwd_intergenic_H3K27ac_5end.bed"

#####read input#####

>rep1 <- read.table(rep1_path,header = F)

>rep2 <- read.table(rep2_path,header = F)

>Dis3$mean_count <- (rep1$V3*Dis3_1_sizeFactor+rep2$V3*Dis3_2_sizeFactor)/rep_num

#####start run#####

>test <- as.matrix(Dis3$mean_count)

>cplist <- c()

>cycle <- length(test)/400

>for(i in 1:cycle-1){

  #print(i)

  seg = segment(test[i*400+1:400*(i+1)], maxk = 400, maxseg = 2)

  cp = unlist(seg@breakpoints)

  #print(cp)

  cplist = c(cplist,cp)

}

>end5 <- data.frame()

>start <- 0

>for(i in cplist){

  index = start*400+i

  end5 = rbind(end5, Dis3[index,])
  start = start+1
}
#####output#####
>write.table(end5,file = output_path, quote = F, sep = "\t",row.names = F,col.names = F)
```

Note: Use start sites of forward strand TUs as an example.

c. Combine start and end sites as refined TUs.

```
>end5 <- read.table("Dis3_fwd_intergenic_H3K27ac_5end.bed",header = F)
>end3 <- read.table("Dis3_fwd_intergenic_H3K27ac_3end.bed",header = F)
>end5$end3 <- end3$V2
>intergenic_H3K27ac_transcripts <- cbind(end5[1:2],end5$end3)
>write.table(intergenic_H3K27ac_transcripts, file = "
Dis3_fwd_intergenic_H3K27ac_refine.bed", quote = F, sep =
"\t",row.names = F, col.names = F)
```

Note: Use forward strand as an example.

30. Give each TU a unique name and remove TUs which overlap with genes.

```
$awk -F ''\t'' '{OFS=''\t''}{print($1,$2,$3,''Dis3_fwd_intergenic_H3K27ac_''NR,''.'',''+'')}' Dis3_fwd_intergenic_H3K27ac_refine.bed > Dis3_fwd_intergenic_H3K27ac_refine2.bed

$awk -F "\t" '{OFS="\t"}{print($1,$2,$3,"Dis3_re_intergenic_H3K27ac_"NR,''.'',"-")}' Dis3_re_intergenic_H3K27ac_refine.bed > Dis3_re_intergenic_H3K27ac_refine2.bed

$bedtools intersect -s -v -a Dis3_fwd_intergenic_H3K27ac_refine2.bed -b FILEPATH/gencode.vM21.chr.annotation.bed |uniq > tmp.bed

$mv tmp.bed Dis3_fwd_intergenic_H3K27ac_refine2.bed

$bedtools intersect -s -v -a Dis3_re_intergenic_H3K27ac_refine2.bed -b FILEPATH/gencode.vM21.chr.annotation.bed |uniq > tmp.bed

$mv tmp.bed Dis3_re_intergenic_H3K27ac_refine2.bed
```

PROMPT and eRNA annotation

Timing: ~1–2 h

Divide candidates of PTs into PROMPTs and eRNAs.

31. TUs, located within 5 kb from 5-'ends of the antisense-strand of protein-coding genes, are selected as PROMPT candidates. Troubleshooting 4.

```
$awk -F "\t" '{OFS="\t"}{match($9,"gene_name \"(.+)\";
level",a)}{if($9~/protein_coding/ &&
$3~/gene/)print($1,$4,$5,a[1],$6,$7)}’
FILEPATH/gencode.vM28.chr_patch_hapl_scaff.annotation.gtf > pcgene.bed
$perl -alne ’{if($F[5] eq "+"){$start=$F[1]-5001;$end=$F[1]-
1;$strand="-"}else{$start=$F[2]+1;$end=$F[2]+5001;$strand="+"}print
join("\t",$F[0],$start,$end,$F[3],0,$strand)}’ pcgene.bed > prompt.bed
$bedtools subtract -s -a prompt.bed -b
FILEPATH/gencode.vM28.chr_patch_hapl_scaff.annotation.bed -A
>prompt_rmgene.bed
$bedtools intersect -wa -a Dis3_fwd_intergenic_H3K27ac_refine2.bed -b
prompt_rmgene.bed -s |uniq> Dis3_fwd_intergenic_PROMPT.bed
$bedtools intersect -wa -a Dis3_re_intergenic_H3K27ac_refine2.bed -b
prompt_rmgene.bed -s |uniq> Dis3_re_intergenic_PROMPT.bed
$cat Dis3_fwd_intergenic_PROMPT.bed Dis3_re_intergenic_PROMPT.bed >
Dis3_intergenic_PROMPT.bed
```

32. TUs, which are located farther than 1 kb upstream of 5′ends and more than 10 kb downstream from 3′ends of protein-coding genes and do not overlap with PROMPT candidates, are selected as eRNA candidates. Troubleshooting 4.

```
$perl -alne ’{if($F[5] eq "+"){$start=$F[1]-1001;$end=$F[1]-1;$strand="+"}else{$start=$F
[2]+1;$end=$F[2]+1001;$strand="-"}print join("\t",$F[0],$start,$end,$F[3],0,$strand)}’
pcgene.bed > TSS1000.bed

$perl -alne ’{if($F[5] eq "+"){$start=$F[2]+1;$end=$F[2]+10001;$strand="+"}else
{$start=$F[1]-1;$end=$F[1]-10001;$strand="-"}print join("\t",$F[0],$start,$end,$F
[3],0,$strand)}’ pcgene.bed > TTS10000.bed

$cat TSS1000.bed TTS10000.bed > gencode.vM21.chr_patch_hapl_scaff.TSS1000TTS10000.bed

$bedtools intersect -v -a Dis3_fwd_intergenic_H3K27ac_refine2.bed -b gencode.vM21.chr_patch_hapl_scaff.TSS1000TTS10000.bed -s |uniq > Dis3_fwd_intergenic_eRNA_candidates.bed

$bedtools intersect -v -a Dis3_re_intergenic_H3K27ac_refine2.bed -b gencode.vM21.chr_patch_hapl_scaff.TSS1000TTS10000.bed -s |uniq > Dis3_re_intergenic_eRNA_candidates.bed

$bedtools intersect -s -v -a Dis3_fwd_intergenic_eRNA_candidates.bed -b Dis3_fwd_intergenic_PROMPT.bed > Dis3_fwd_intergenic_eRNA.bed

$bedtools intersect -s -v -a Dis3_re_intergenic_eRNA_candidates.bed -b Dis3_re_intergenic_PROMPT.bed > Dis3_re_intergenic_eRNA.bed

$cat Dis3_fwd_intergenic_eRNA.bed Dis3_re_intergenic_eRNA.bed > Dis3_intergenic_eRNA.bed
```

33. Calculate the counts of coding genes, eRNAs and PROMPTs.

```
$cat Dis3_intergenic_eRNA.bed Dis3_intergenic_PROMPT.bed | awk -F "\t" ’{OFS="\t"}{print($1,".","exon",$2,$3,".",$6,".","gene_id \""$4"\";")}’ > eRNA_PRMOPT.gtf

$cat eRNA_PRMOPT.gtf FILEPATH/ gencode.vM28.chr_patch_hapl_scaff.annotation.gtf >
gene_eRNA_PRMOPT.gtf

$featureCounts -p -g gene_name -a gene_eRNA_PRMOPT.gtf -s 2 -o mm10_Dis3_RNAseqcount.txt
sg4841_mm10_filter.bam sg9270_mm10_filter.bam sgDis3_1_mm10_filter.bam sgDis3_2_mm10_
filter.bam
```

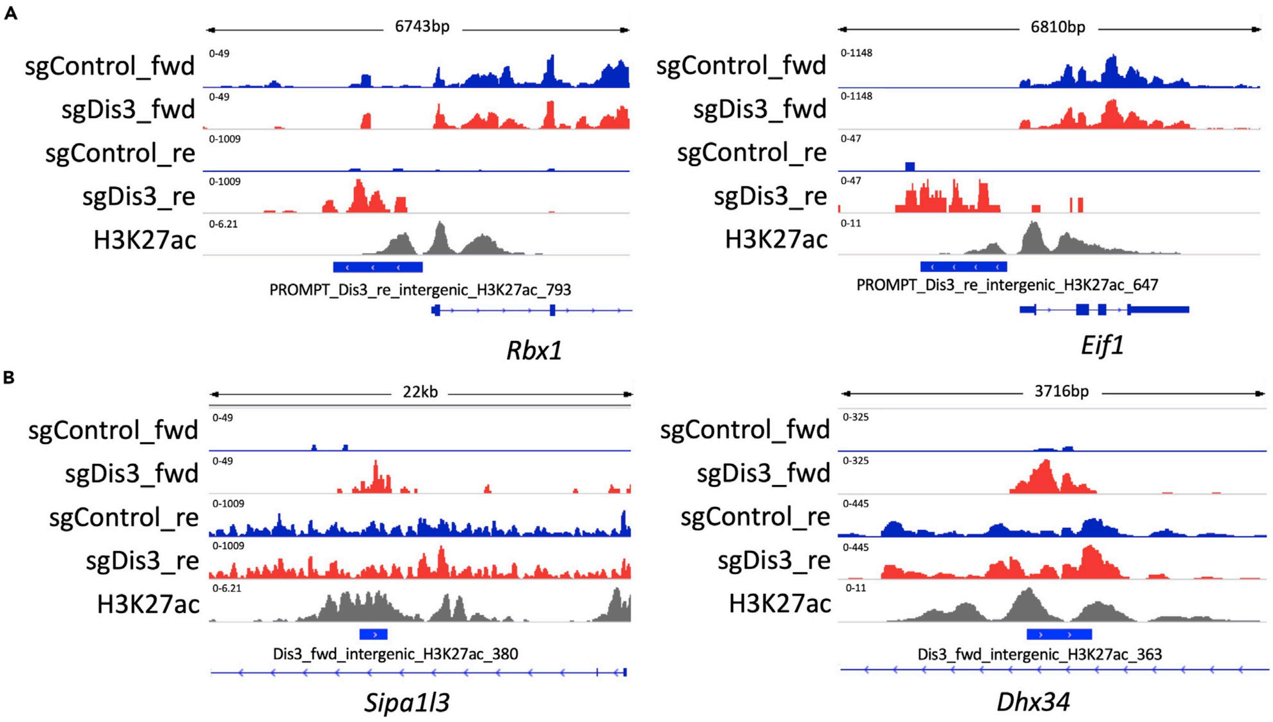


Figure 2. Examples of PT annotations

(A) Blue boxes in bottom track showed the annotations of *Pbx1* and *Eif1* PROMPTs.

(B) Blue boxes in bottom track showed the annotations of eRNAs next to *Sipa1l3* and *Dhx34*.

34. Use edgeR to perform the differential expression analyses of eRNAs and PROMPTs.

```
>library(edgeR)

>data <- read.table(file = "mm10_Dis3_RNAseqcount.txt", sep = "\t", header = T,skip =1)

>sampleNames <- c("C_1","C_2","t_1","t_2")

>names(data)[7:10] <- sampleNames

>head(data)

>countMatrix <- as.matrix(cbind(data[7:10]))

>rownames(countMatrix) <- data$Geneid

>head(countMatrix)

>group <- factor(c("c","c","t","t"))

>y <- DGEList(counts=countMatrix,group=group)

>y

#####FILTER#####

>keep <- rowSums(cpm(y)>0) >= 2;

>y <- y[keep,keep.lib.sizes=FALSE];

>dim(y)

#####dm6 spike-in normalization#####
```

```
>dm6Dis3KD <- c(0.9232670, 1.1442665, 0.9295284, 1.0183166)

>y$samples$norm.factors <- dm6Dis3KD

>plotMDS(y)

>design <- model.matrix(~group)

>rownames(design) <- colnames(y)

>y <- estimateDisp(y,design)

>plotBCV(y);

>fit <- glmQLFit(y,design)

>qlf <- glmQLFTest(fit,coef=2)

>topTags(qlf)

>top <- rownames(topTags(qlf))

>summary(de <- decideTestsDGE(qlf))

>detags <- rownames(y)[as.logical(de)];

>plotSmear(qlf, de.tags=detags);

>abline(h=c(-1, 1), col="blue");

>cpm <- cpm(y)[rownames(y),]

>tmp <- qlf$table

>tmp <- cbind(tmp,cpm)

>exp <- cbind(data[2:10])

>exp$gene <- data[,1]

>tmp$gene <- rownames(y)

>tmp <- merge(tmp,exp,by="gene")

>write.csv(tmp,file = "Dis3KD_RNAseq.csv")
```

Note: "dm6Dis3KD" is a vector of sizeFactors calculated in step 23.

35. Select eRNAs and PROMPTs whose both p values are less than 0.05 and whose $\log_2$fold change (fold change, FC, is sgDis3 divided by sgControl) are greater than 1 as the final eRNAs and PROMPTs respectively.
36. Use the final eRNAs and PROMPTs to make BED files named "Dis3up_eRNA.bed" and "Dis3up_PROMPT.bed". Troubleshooting 5.

Quantify pervasive transcripts

Timing: ~10 min

Use the median length to quantify PTs.

37. Calculate the median length of the final PROMPTs and eRNAs. The median length is around 1.6 kb.

EXPECTED OUTCOMES

This protocol inputs the data of *Dis3* depletion chromatin RNA-seq and outputs the coordinates of PTs. The coordinates of eRNAs and PROMPTs stabilized by *Dis3* depletion are annotated in the ''Dis3up_eRNA.bed'' and ''Dis3up_PROMPT.bed'' BED files generated in step 36. The average length of PTs is around 1.6 kb. The annotated PTs can be used to calculate their change of expression and location in different treatments. The annotations of PROMPTs of *Rbx1* and *Eif1* are shown in Figure 2A and those of eRNAs next to *Sipa1l3* and *Dhx34* are shown in Figure 2B.

LIMITATIONS

This protocol aims to identify PTs de novo. To increase genomic coverage and signal of PTs, high quality chromatin-associated RNA and appropriate depth of sequencing are needed. The higher signals PTs have, the more accurate annotation of their coordinates will be. The main limitation of this protocol is that it cannot identify PTs that are in the same orientation as expressed protein coding genes due to the higher reads from mRNAs. Additionally, this protocol only identifies PTs that are normally degraded by the nuclear exosome complex. We compared PTs identified by TT-seq and chromatin RNA-seq upon *Dis3* depletion respectively. The PTs identified by these two approaches mostly overlapped but TT-seq can identify shorter PTs. This is due to the limitation of the library preparation in chromatin RNA-seq which removes most RNAs smaller than 100 nt. Meanwhile, incorporation of additional sequencing data such as H3K4me1 ChIP-seq and PRO-seq can provide higher accuracy of PTs definition.

TROUBLESHOOTING

Problem 1

Low yield of RNA extraction of chromatin associated RNA may arise in step 10.

Potential solution

- Increase the starting materials (e.g., higher number of cells).
- Adjust the amount of TRIzol (for RNA extraction) to avoid incomplete cell lysis due to the insufficient amount of TRIzol.
- Verify all materials and reagents are RNase free.

Problem 2

RNA with low A260/280 and A260/230 ratios may occur in step 10. Low RIN values should be expected in step 9.

Potential solution

- Repeat RNA precipitation step if low A260/280 and A260/230 ratios.
- Low RIN value expected for chromatin RNA samples because 18s and 28s rRNAs are located in the cytoplasm.

Problem 3

Low mapping reads of dm6 may occur in step 21.

Potential solution

- Increase the percentage of spike-in control S2 cells.

Problem 4

BED files have blank Lines may occur in steps 27, 28, 31, and 32.

Potential solution

- Delete these blank lines in vim state by using "%s/^M//g".

Problem 5

Low annotation number of PTs may occur in step 36.

Potential solution

- Loosen the cutoff (such as $Log_2FC>0$ and $p<0.05$) to include more upregulated PTs upon *Dis3* depletion.

RESOURCE AVAILABILITY

Lead contact

Further information and requests for resources and reagents should be directed to the lead contact, Mo Chen (mochen@mail.tsinghua.edu.cn).

Materials availability

This protocol did not generate new unique reagents.

Data and code availability

The data used in this study are described in Liu et al. (2022). The GEO number of RNA-seq data is GEO: GSE162829. The GEO number of ChIP-seq data is GEO: GSE162842. This protocol includes all codes.

ACKNOWLEDGMENTS

M.C. received start-up support from Tsinghua University, the Peking University-Tsinghua Center for Life Sciences. M.C. is also supported by Natural Science Foundation of China (grant numbers 81870122 and 92068114) and National Key R&D Program of China (grant number 2021YFA1300100).

AUTHOR CONTRIBUTIONS

M.C. conceived and supervised the project. M.C. and X.L. designed the experiments. X.L. performed the experiments. Z.G. performed bioinformatics analyses and wrote the protocol.

DECLARATION OF INTERESTS

The authors declare no competing interests.

REFERENCES

David, C.J., Huang, Y.H., Chen, M., Su, J., Zou, Y., Bardeesy, N., Iacobuzio- Donahue, C.A., and Massague, J. (2016). TGF-beta tumor suppression through a lethal EMT. Cell *164*, 1015–1030. https://doi.org/10.1016/j.cell.2016.01.009.

Dobin, A., and Gingeras, T.R. (2015). Mapping RNA-seq reads with STAR. Curr. Protoc. Bioinformatics *51*, 11.14.11–11.14.19. https://doi.org/10.1002/0471250953.bi1114s51.

Doench, J.G., Fusi, N., Sullender, M., Hegde, M., Vaimberg, E.W., Donovan, K.F., Smith, I., Tothova, Z., Wilen, C., Orchard, R., et al. (2016). Optimized sgRNA design to maximize activity and minimize off-target effects of CRISPR-Cas9. Nat. Biotechnol. *34*, 184–191. https://doi.org/10.1038/nbt.3437.

Horlbeck, M.A., Gilbert, L.A., Villalta, J.E., Adamson, B., Pak, R.A., Chen, Y., Fields, A.P., Park, C.Y., Corn, J.E., Kampmann, M., and Weissman, J.S. (2016). Compact and highly active next-generation libraries for CRISPR-mediated gene repression and activation. Elife *5*, e19760. https://doi.org/10.7554/elife.19760.

Huber, W., Toedling, J., and Steinmetz, L.M. (2006). Transcript mapping with high-density oligonucleotide tiling arrays. Bioinformatics *22*, 1963–1970. https://doi.org/10.1093/bioinformatics/btl289.

Langmead, B., and Salzberg, S.L. (2012). Fast gapped-read alignment with Bowtie 2. Nat. Methods *9*, 357–359. https://doi.org/10.1038/nmeth.1923.

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., Marth, G., Abecasis, G., Durbin, R., and Genome Project Data Processing, S. (2009). The sequence alignment/map format and SAMtools. Bioinformatics *25*, 2078–2079. https://doi.org/10.1093/bioinformatics/btp352.

Liao, Y., Smyth, G.K., and Shi, W. (2014). featureCounts: an efficient general purpose program for assigning sequence reads to genomic features. Bioinformatics *30*, 923–930. https://doi.org/10.1093/bioinformatics/btt656.

Liu, X., Guo, Z., Han, J., Peng, B., Zhang, B., Li, H., Hu, X., David, C.J., and Chen, M. (2022). The PAF1 complex promotes 3′ processing of pervasive transcripts. Cell Rep. *38*, 110519. https://doi.org/10.1016/j.celrep.2022.110519.

Quinlan, A.R., and Hall, I.M. (2010). BEDTools: a flexible suite of utilities for comparing genomic features. Bioinformatics *26*, 841–842. https://doi.org/10.1093/bioinformatics/btq033.

Ramirez, F., Ryan, D.P., Gruning, B., Bhardwaj, V., Kilpert, F., Richter, A.S., Heyne, S., Dundar, F., and Manke, T. (2016). deepTools2: a next generation web server for deep-sequencing data analysis. Nucleic Acids Res. *44*, W160–W165. https://doi.org/10.1093/nar/gkw257.

Robinson, M.D., McCarthy, D.J., and Smyth, G.K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. Bioinformatics *26*, 139–140. https://doi.org/10.1093/bioinformatics/btp616.

Stojic, L., Lun, A.T.L., Mangei, J., Mascalchi, P., Quarantotti, V., Barr, A.R., Bakal, C., Marioni, J.C., Gergely, F., and Odom, D.T. (2018). Specificity of RNAi, LNA and CRISPRi as loss-of-function methods in transcriptional analysis. Nucleic Acids Res. *46*, 5950–5966. https://doi.org/10.1093/nar/gky437.

Zacher, B., Michel, M., Schwalb, B., Cramer, P., Tresch, A., and Gagneur, J. (2017). Accurate promoter and enhancer identification in 127 ENCODE and roadmap epigenomics cell types and tissues by GenoSTAN. PLoS One *12*, e0169249. https://doi.org/10.1371/journal.pone.0169249.

Zhang, Y., Liu, T., Meyer, C.A., Eeckhoute, J., Johnson, D.S., Bernstein, B.E., Nusbaum, C., Myers, R.M., Brown, M., Li, W., and Liu, X.S. (2008). Model-based analysis of ChIP-seq (MACS). Genome Biol. *9*, R137. https://doi.org/10.1186/gb-2008-9-9-r137.