nature protocols

https://doi.org/10.1038/s41596-023-00898-5

Protocol

Check for updates

EpiTyping: analysis of epigenetic aberrations in parental imprinting and X-chromosome inactivation using RNA-seq

In the format provided by the authors and unedited

Supplementary Table 1 - File names

File name	File description	Step
Epityping.git	GIT repository	1
GRCh38.primary_assembly.genome.fa	FASTA genome file-human	2
gencode.v42.primary_assembly.annotation.gtf	GTF annotation file-human	2
GRCm39.primary_assembly.genome.fa	FASTA genome file-mouse	2
gencode.vM31.primary_assembly.annotation.gtf	GTF annotation file-mouse	2
GCF_000001405.39	VCF dbSNP file	3
dbSNPbuild155Renamed.vcf.gz	Renamed zipped dbSNP VCF file	4
GRCh38_exome.bed.gz	Zipped BED file of exon coordinates	5

Supplementary Table 1-File names

The names of the files in the Set-Up stages, along with their description and corresponding steps.

Supplementary Table 2 - Samples for testing

SRA	Bioproject	Library Lay	Cell line	Naïve/Prim	RNA or DN	Sex
ERR3466738	PRJEB3387	PAIRED	H1	Naïve	RNA	Male
ERR3466740	PRJEB3387	PAIRED	H1	Primed	RNA	Male
SRR4301709	PRJNA3447	SINGLE	H9	Naïve	RNA	Female
SRR1028755	PRJNA2273	PAIRED	H9	Primed	RNA	Female
SRR3933201	PRJNA3307	PAIRED	H9	Primed	RNA	Female
SRR2070629	PRJNA2871	PAIRED	H1		DNA	Male

Supplementary Table 2-Samples for testing

Accession numbers and additional information of sequencing samples for optional further testing.

SUPPLEMENTARY FILE 1 - MAIN

####DATA SETUP###

```
cd  ~
git clone https://github.com/Gal-Keshet/EpiTyping.git

cd ~/EpiTyping/genome_files
mkdir star_ref
cd ~/EpiTyping/genome_files/star_ref

##Human genome
wget
https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_42/GRCh
38.primary_assembly.genome.fa.gz
gunzip  GRCh38.primary_assembly.genome.fa.gz
wget
https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_human/release_42/genc
ode.v42.primary_assembly.annotation.gtf.gz
gunzip  gencode.v42.primary_assembly.annotation.gtf.gz


##Mouse genome
wget
https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_mouse/release_M31/GRC
m39.primary_assembly.genome.fa.gz
gunzip  GRCm39.primary_assembly.genome.fa.gz
wget
https://ftp.ebi.ac.uk/pub/databases/gencode/Gencode_mouse/release_M31/gen
code.vM31.primary_assembly.annotation.gtf.gz
gunzip  gencode.vM31.primary_assembly.annotation.gtf.gz

#snp files
cd ~/EpiTyping/genome_files
wget  https://ftp.ncbi.nih.gov/snp/archive/b155/VCF/GCF_000001405.39.gz
wget
https://ftp.ncbi.nih.gov/snp/archive/b155/VCF/GCF_000001405.39.gz.tbi

bcftools annotate --threads 10 --output-type z --rename-chrs
remapNCBI.txt --output dbSNPbuild155Renamed.vcf.gz GCF_000001405.38.gz
tabix dbSNPbuild155Renamed.vcf.gz
rm GCF_000001405.39.gz

##STAR index-human
STAR --runThreadN 10 --runMode genomeGenerate --genomeDir
~/EpiTyping/genome_files/star_index --genomeFastaFiles
~/EpiTyping/genome_files/star_ref/GRCh38.primary_assembly.genome.fa --
sjdbGTFfile
~/EpiTyping/genome_files/star_ref/gencode.v42.primary_assembly.annotation
.gtf --sjdbOverhang 100
##STAR index-mouse
STAR --runThreadN 10 --runMode genomeGenerate --genomeDir
~/EpiTyping/genome_files/star_mouse_index --genomeFastaFiles
~/EpiTyping/genome_files/star_ref/GRCm39.primary_assembly.genome.fa --
```

```
sjdbGTFfile
~/EpiTyping/genome_files/star_ref/gencode.vM31.primary_assembly.annotatio
n.gtf --sjdbOverhang 100

##GATK dictionary
cd ~/EpiTyping/genome_files/star_ref
samtools faidx GRCh38.primary_assembly.genome.fa
gatk CreateSequenceDictionary -R GRCh38.primary_assembly.genome.fa

##Exon ccordinates BED file
cd ~/EpiTyping/genome_files
awk '{if($3=="exon") {print $1"\t"$4-
1"\t"$5+1"\t"substr($16,2,length($16)-3)}}'
star_ref/gencode.v42.primary_assembly.annotation.gtf | sort -k 1,1 -k2,2n
| bgzip > GRCh38_exome.bed.gz
tabix GRCh38_exome.bed.gz

##BWA index of human and mouse genomes - for DNA integration (LOI
analysis)

cd ~/EpiTyping/genome_files/star_ref
bwa index GRCh38.primary_assembly.genome.fa
bwa index GRCm39.primary_assembly.genome.fa

###EXPIRAMENTAL PROCEDURE###

##Aquire sample

prefetch ERR3466738 -O ~/EpiTyping/fastq/
fastq-dump --split-files ~/EpiTyping/fastq/ERR3466738/ERR3466738.sra -O
~/EpiTyping/fastq/ --gzip
SAMPLE=ERR3466738

##Assigning variables

PROJECT_DIR="~/EpiTyping"
REF_GENOME="~/EpiTyping/genome_files/star_ref/GRCh38.primary_assembly.gen
ome.fa"
STAR_INDEX="~/EpiTyping/genome_files/star_index"
STAR_MOUSE_INDEX="~/EpiTyping/genome_files/star_mouse_index"
DBSNP="~/EpiTyping/genome_files/dbSNPbuild155Renamed.vcf.gz"
GTF="~/EpiTyping/genome_files/star_ref/gencode.v42.primary_assembly.annot
ation.gtf"
INTERVAL_LIST="~/EpiTyping/genome_files/GRCh38_exome.bed.gz"
IMPRINTED_INTERVAL_LIST="~/EpiTyping/genome_files/imprinted_genes_sorted.
bed"
HEADER="~/EpiTyping/genome_files/headers_for_annotation.txt"
ADAPTERS="~/EpiTyping/genome_files/CommonAdapters.fa"
THREADS=10

##Making relevant directories

mkdir ${PROJECT_DIR}/star_out
mkdir ${PROJECT_DIR}/vcf
```

```
mkdir ${PROJECT_DIR}/bam
mkdir ${PROJECT_DIR}/star_mouse_out
mkdir ${PROJECT_DIR}/feature_counts

##Trimming the fastq file

#PAIRED library layout
java -jar trimmomatic-0.39.jar PE -threads ${THREADS}
${PROJECT_DIR}/fastq/${SAMPLE}_1.fastq.gz
${PROJECT_DIR}/fastq/${SAMPLE}_2.fastq.gz
${PROJECT_DIR}/fastq/${SAMPLE}_trimmed_1.fastq.gz
${PROJECT_DIR}/fastq/${SAMPLE}_trimmed_1_unpaired.fastq.gz
${PROJECT_DIR}/fastq/${SAMPLE}_trimmed_2.fastq.gz
${PROJECT_DIR}/fastq/${SAMPLE}_trimmed_2_unpaired.fastq.gz
ILLUMINACLIP:${ADAPTERS}:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15
MINLEN:36
#SINGLE library layout
java -jar trimmomatic-0.39.jar SE -threads ${THREADS}
${PROJECT_DIR}/fastq/${SAMPLE}.fastq.gz
${PROJECT_DIR}/fastq/${SAMPLE}_trimmed.fastq.gz
ILLUMINACLIP:${ADAPTERS}:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15
MINLEN:36

##Alignment using STAR-human

#PAIRED library layout:
STAR --runThreadN ${THREADS}\
 --genomeDir ${STAR_INDEX}\
 --readFilesIn ${PROJECT_DIR}/fastq/${SAMPLE}_trimmed_1.fastq.gz
${PROJECT_DIR}/fastq/${SAMPLE}_trimmed_2.fastq.gz\
 --outSAMtype BAM SortedByCoordinate\
 --twopassMode Basic\
 --outSAMattributes NM\
 --readFilesCommand zcat\
 --limitBAMsortRAM 35145460689\
 --outFileNamePrefix ${PROJECT_DIR}/star_out/${SAMPLE}_

#SINGLE library layout:
STAR --runThreadN ${THREADS}\
 --genomeDir ${STAR_INDEX}\
 --readFilesIn ${PROJECT_DIR}/fastq/${SAMPLE}_trimmed.fastq.gz\
 --outSAMtype BAM SortedByCoordinate\
 --twopassMode Basic\
 --outSAMattributes NM\
 --readFilesCommand zcat\
 --limitBAMsortRAM 35145460689\
 --outFileNamePrefix ${PROJECT_DIR}/star_out/${SAMPLE}_

##Alignment unsing STAR-mouse

#PAIRED library layout
STAR --runThreadN ${THREADS}\
 --genomeDir ${STAR_MOUSE_INDEX}\
```

```
 --readFilesIn ${PROJECT_DIR}/fastq/${SAMPLE}_trimmed_1.fastq.gz
${PROJECT_DIR}/fastq/${SAMPLE}_trimmed_2.fastq.gz\
 --outSAMtype BAM SortedByCoordinate\
 --twopassMode Basic\
 --outSAMattributes NM\
 --readFilesCommand zcat\
 --limitBAMsortRAM 35145460689\
 --outFileNamePrefix ${PROJECT_DIR}/star_mouse_out/${SAMPLE}_mouse_

#SINGLE library layout
STAR --runThreadN ${THREADS}\
 --genomeDir ${STAR_MOUSE_INDEX}\
 --readFilesIn ${PROJECT_DIR}/fastq/${SAMPLE}_trimmed.fastq.gz\
 --outSAMtype BAM SortedByCoordinate\
 --twopassMode Basic\
 --outSAMattributes NM\
 --readFilesCommand zcat\
 --limitBAMsortRAM 35145460689\
 --outFileNamePrefix ${PROJECT_DIR}/star_mouse_out/${SAMPLE}_mouse_

##Executing Xenofilterr.R script

human=${PROJECT_DIR}/star_out/${SAMPLE}_Aligned.sortedByCoord.out.bam
mouse=${PROJECT_DIR}/star_mouse_out/${SAMPLE}_mouse_Aligned.sortedByCoord
.out.bam
destination=${PROJECT_DIR}/bam

Rscript ./xenofilterr.R ${human} ${mouse} ${destination}


##Generating a count table

#PAIRED library layout:
featureCounts\
 --extraAttributes "gene_name"\
 -T ${THREADS}\
 -a ${GTF}\
 -s 0\ # (0: unstranded, 1: forword, 2: rev)
 -o ${PROJECT_DIR}/feature_counts/${SAMPLE}_counts.txt\
 -p\
 --countReadPairs\
${PROJECT_DIR}/bam/Filtered_bams/${SAMPLE}_Aligned.sortedByCoord.out_Filt
ered.bam

tail -n +3 ${PROJECT_DIR}/feature_counts/${SAMPLE}_counts.txt | awk
'BEGIN { print "Gene ID\tChr\tStart\tLength\tsymbol\tcounts" } { print $1
"\t" $2 "\t" $3 "\t" $6 "\t" $7 "\t" $8 }' >
${PROJECT_DIR}/feature_counts/${SAMPLE}_edited_counts.txt

#SINGLE library layout:
featureCounts\
      --extraAttributes "gene_name"\
      -T ${THREADS}\
      -a ${GTF}\\
```

```
      -s 0\ # (0: unstranded, 1: forword, 2: rev)
      -o ${PROJECT_DIR}/feature_counts/${SAMPLE}_counts.txt\
${PROJECT_DIR}/bam/Filtered_bams/${SAMPLE}_Aligned.sortedByCoord.out_Filt
ered.bam

tail -n +3 ${PROJECT_DIR}/feature_counts/${SAMPLE}_counts.txt | awk
'BEGIN { print "Gene ID\tChr\tStart\tLength\tsymbol\tcounts" } { print $1
"\t" $2 "\t" $3 "\t" $6 "\t" $7 "\t" $8 }' >
${PROJECT_DIR}/feature_counts/${SAMPLE}_edited_counts.txt

##Adding reads groups

java -jar picard.jar AddOrReplaceReadGroups\

I=${PROJECT_DIR}/bam/Filtered_bams/${SAMPLE}_Aligned.sortedByCoord.out_Fi
ltered.bam\
 O=${PROJECT_DIR}/bam/${SAMPLE}_Aligned.out_rg.bam\
 SO=coordinate RGID=rnasq RGLB=lb RGPL=illumina RGPU=pu RGSM=${SAMPLE}

##Marking the duplicated reads of the file

java -jar picard.jar MarkDuplicates\
 I=${PROJECT_DIR}/bam/${SAMPLE}_Aligned.out_rg.bam\
 O=${PROJECT_DIR}/bam/${SAMPLE}_dedupped.bam\
 CREATE_INDEX=true VALIDATION_STRINGENCY=SILENT M=output.metrics


##Modifying N-containing reads

gatk SplitNCigarReads\
 -R ${REF_GENOME}\
 -I ${PROJECT_DIR}/bam/${SAMPLE}_dedupped.bam\
 -O ${PROJECT_DIR}/bam/${SAMPLE}_split.bam

##Base recalibrating

gatk BaseRecalibrator\
 -L ${INTERVAL_LIST}\
 -ip 100\
 -I ${PROJECT_DIR}/bam/${SAMPLE}_split.bam\
 -R ${REF_GENOME}\
 --known-sites ${DBSNP}\
 -O ${PROJECT_DIR}/bam/${SAMPLE}_recal_data.table

gatk ApplyBQSR\
 -R ${REF_GENOME}\
 -I ${PROJECT_DIR}/bam/${SAMPLE}_split.bam\
 -L ${INTERVAL_LIST}\
 -ip 100\
 --bqsr-recal-file ${PROJECT_DIR}/bam/${SAMPLE}_recal_data.table\
 -O ${PROJECT_DIR}/bam/${SAMPLE}_recalibrated.bam

##Calling genomic variants
```

```
gatk --java-options "-Xmx8G" HaplotypeCaller\
 -R ${REF_GENOME}\
 -I ${PROJECT_DIR}/bam/${SAMPLE}_recalibrated.bam\
 -L ${INTERVAL_LIST}\
 --dont-use-soft-clipped-bases\
 -stand-call-conf 20.0\
 -O ${PROJECT_DIR}/vcf/${SAMPLE}_output.vcf

##Filtering variants VCF file

gatk VariantFiltration\
 -R ${REF_GENOME}\
 -V ${PROJECT_DIR}/vcf/${SAMPLE}_output.vcf\
 -O ${PROJECT_DIR}/vcf/${SAMPLE}_filtered.vcf\
 -L ${INTERVAL_LIST}\
 -window 35 -cluster 3\
 --filter-name FS --filter-expression "FS > 30.0"\
 --filter-name QD --filter-expression "QD < 2.0"

##Naming and indexing

bgzip ${PROJECT_DIR}/vcf/${SAMPLE}_filtered.vcf
tabix ${PROJECT_DIR}/vcf/${SAMPLE}_filtered.vcf.gz

bcftools annotate\
 --threads $THREADS\
 --annotations ${INTERVAL_LIST}\
 -h $HEADER\
 -c CHROM,FROM,TO,Gene\
 --output-type z\
 --output ${PROJECT_DIR}/vcf/${SAMPLE}_with_names.vcf.gz\
 ${PROJECT_DIR}/vcf/${SAMPLE}_filtered.vcf.gz

tabix ${PROJECT_DIR}/vcf/${SAMPLE}_with_names.vcf.gz

#Annotating VCF file

bcftools annotate\
 --threads $THREADS\
 -a ${DBSNP}\
 -c ID,INFO/RS,INFO/COMMON\
 --output-type z\
 --output ${PROJECT_DIR}/vcf/${SAMPLE}_dbSNP.vcf.gz\
 ${PROJECT_DIR}/vcf/${SAMPLE}_with_names.vcf.gz

tabix ${PROJECT_DIR}/vcf/${SAMPLE}_dbSNP.vcf.gz

##Further filtering of variants

bcftools view\
 --threads $THREADS\
 -i 'INFO/RS!="." & FILTER="PASS" & FMT/DP > 9'\
 --output-type z\
 --output-file ${PROJECT_DIR}/vcf/${SAMPLE}_dbSNP_filtered.vcf.gz\
```

```
 ${PROJECT_DIR}/vcf/${SAMPLE}_dbSNP.vcf.gz

tabix ${PROJECT_DIR}/vcf/${SAMPLE}_dbSNP_filtered.vcf.gz

##Creating variant table - X inactivation analysis

bcftools query\
 -H -f
'%CHROM\t%POS\t%ID\t%INFO/Gene\t%INFO/COMMON\t%REF\t%ALT[\t%GT:%AD{0}:%AD
{1}:%DP]\n'\
 --output ${PROJECT_DIR}/vcf/xci_final_table.txt\
 ${PROJECT_DIR}/vcf/${SAMPLE}_dbSNP_filtered.vcf.gz

##Creating variant table - LOI analysis

bcftools query\
 -H -f
'%CHROM\t%POS\t%ID\t%INFO/Gene\t%INFO/COMMON\t%REF\t%ALT[\t%GT:%AD{0}:%AD
{1}:%DP]\n'\
 --regions-file ${IMPRINTED_INTERVAL_LIST}\
 --output ${PROJECT_DIR}/vcf/loi_final_table.txt\
 ${PROJECT_DIR}/vcf/${SAMPLE}_dbSNP_filtered.vcf.gz

###DNA INTEGRATION###


##Aquiring DNA sample

prefetch SRR2070629 --max-size 200g -O ~/EpiTyping/dna_fastq/
fastq-dump --split-files ~/EpiTyping/dna_fastq/SRR2070629/SRR2070629.sra
-O ~/EpiTyping/dna_fastq --gzip
DNA_SAMPLE=SRR2070629
dna_fastq_FOLDER="${PROJECT_DIR}/dna_fastq/"
DNA_REF="~/EpiTyping/genome_files/star_ref/GRCh38.primary_assembly.genome
.fa"
DNA_MOUSE_REF="~/EpiTyping/genome_files/star_ref/GRCm39.primary_assembly.
genome.fa"

##Trimming DNA fastq

#PAIRED library layout
java -jar trimmomatic-0.39.jar PE -threads ${THREADS}
${PROJECT_DIR}/dna_fastq/${DNA_SAMPLE}_1.fastq.gz
${PROJECT_DIR}/dna_fastq/${DNA_SAMPLE}_2.fastq.gz
${PROJECT_DIR}/dna_fastq/${DNA_SAMPLE}_trimmed_1.fastq.gz
${PROJECT_DIR}/dna_fastq/${DNA_SAMPLE}_trimmed_1_unpaired.fastq.gz
${PROJECT_DIR}/dna_fastq/${DNA_SAMPLE}_trimmed_2.fastq.gz
${PROJECT_DIR}/dna_fastq/${DNA_SAMPLE}_trimmed_2_unpaired.fastq.gz
ILLUMINACLIP:${ADAPTERS}:2:30:10 LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15
MINLEN:36
#SINGLE library layout
java -jar trimmomatic-0.39.jar SE -threads ${THREADS}
${PROJECT_DIR}/dna_fastq/${DNA_SAMPLE}.fastq.gz
```

```
${DNA_SAMPLE}_trimmed.fastq.gz  ILLUMINACLIP:${ADAPTERS}:2:30:10
LEADING:3 TRAILING:3 SLIDINGWINDOW:4:15 MINLEN:36

##Aligning DNA sample to human reference genome using BWA

mkdir ${PROJECT_DIR}/DNA_BAM
DNA_BAM_FOLDER="${PROJECT_DIR}/DNA_BAM/"

#PAIRED library layout

bwa mem -M -t ${THREADS}\
 ${DNA_REF}\
 ${dna_fastq_FOLDER}/${DNA_SAMPLE}_trimmed_1.fastq.gz
${dna_fastq_FOLDER}/${DNA_SAMPLE}_trimmed_2.fastq.gz\
 | samtools view -b -F 4 -@ ${THREADS} -o
${DNA_BAM_FOLDER}/${DNA_SAMPLE}_aligned.bam

#SINGLE library layout

bwa mem -M -t ${THREADS}\
 ${DNA_REF}\
 ${dna_fastq_FOLDER}/${DNA_SAMPLE}_trimmed.fastq
 | samtools view -b -F 4 -@ ${THREADS} -o
${DNA_BAM_FOLDER}/${DNA_SAMPLE}_aligned.bam

##Aligning DNA sample to mouse reference genome using BWA

#PAIRED library layout

bwa mem -M -t ${THREADS}\
 ${DNA_MOUSE_REF}\
 ${dna_fastq_FOLDER}/${DNA_SAMPLE}_trimmed_1.fastq.gz
${dna_fastq_FOLDER}/${DNA_SAMPLE}_trimmed_2.fastq.gz\
 | samtools view -b -F 4 -@ ${THREADS} -o
${DNA_BAM_FOLDER}/${DNA_SAMPLE}_mouse_aligned.bam

#SINGLE library layout

bwa mem -M -t ${THREADS}\
 ${DNA_MOUSE_REF}\
 ${dna_fastq_FOLDER}/${DNA_SAMPLE}.trimmed.fastq
 | samtools view -b -F 4 -@ ${THREADS} -o
${DNA_BAM_FOLDER}/${DNA_SAMPLE}_mouse_aligned.bam

##Sorting mouse BAM file

mkdir ${DNA_BAM_FOLDER}/PROCESSED

java -jar picard.jar SortSam\
 I=${DNA_BAM_FOLDER}/${DNA_SAMPLE}_mouse_aligned.bam\
 O=${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_mouse_aligned_sorted.bam\
 SORT_ORDER=coordinate CREATE_INDEX=true

##Cutting human BAM file
```

```
samtools view -b -L ${INTERVAL_LIST} -o
${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_cut.bam -@ ${THREADS}
${DNA_BAM_FOLDER}/${DNA_SAMPLE}_aligned.bam

##Adding read groups

java -jar picard.jar AddOrReplaceReadGroups\
 I=${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_cut.bam\
 O=${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_Aligned.out_rg.bam\
 SO=coordinate RGID=dnasq RGLB=lb RGPL=illumina RGPU=pu\
 RGSM=${DNA_SAMPLE}

##Sorting the file
java -jar picard.jar SortSam\
 I=${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_Aligned.out_rg.bam\
 O=${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_Aligned.out_rg_sorted.bam\
 SORT_ORDER=coordinate CREATE_INDEX=true

##Using XenoFilteR to filter out mouse-aligned reads

human_dna=${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_Aligned.out_rg_sorted
.bam
mouse_dna=${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_mouse_aligned_sorted.
bam
destination_dna=${DNA_BAM_FOLDER}

Rscript ./xenofilterr.R ${human_dna} ${mouse_dna} ${destination_dna}

##Marking the duplicated reads

java -jar picard.jar MarkDuplicates\
 -I
${DNA_BAM_FOLDER}/Filtered_bams/${DNA_SAMPLE}_Aligned.out_rg_sorted_Filte
red.bam\
 -O ${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_dedupped.bam\
 -CREATE_INDEX true -VALIDATION_STRINGENCY SILENT -M output.metrics

##Base recalibrating

gatk BaseRecalibrator\
 -L ${INTERVAL_LIST}\
 -ip 100\
 -I ${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_dedupped.bam\
 -R ${DNA_REF}\
 --known-sites ${DBSNP}\
 -O ${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_recal_data.table

gatk ApplyBQSR\
 -R ${DNA_REF}\
 -I ${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_dedupped.bam\
 -L ${INTERVAL_LIST}\
 -ip 100\
```

```
 --bqsr-recal-file
${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_recal_data.table\
 -O ${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_recalibrated.bam

samtools index ${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_recalibrated.bam

##Calling genomic variants

mkdir ${PROJECT_DIR}/DNA_VCF
gatk --java-options "-Xmx8G" HaplotypeCaller\
 -R ${DNA_REF}\
 -I ${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_recalibrated.bam\
 -L ${INTERVAL_LIST}\
 --dont-use-soft-clipped-bases\
 -stand-call-conf 20.0\
 -bamout ${DNA_BAM_FOLDER}/PROCESSED/${DNA_SAMPLE}_bam_out.bam\
 -O ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_output.vcf

bgzip ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_output.vcf
tabix ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_output.vcf.gz

##Filtering variants

gatk --java-options "-Xms3000m" VariantFiltration\
 -R ${DNA_REF}\
 -V ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_output.vcf.gz\
 -O ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_filtered.vcf\
 -L ${INTERVAL_LIST}\
 --filter-expression "QD < 2.0 || FS > 30.0 || SOR > 3.0 || MQ < 40.0 ||
MQRankSum < -3.0 || ReadPosRankSum < -3.0"\
 --filter-name "HardFiltered"
bgzip ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_filtered.vcf
tabix ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_filtered.vcf.gz

##Annotating variants VCF file

gatk --java-options -Xmx10g CNNScoreVariants\
 -V ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_filtered.vcf.gz\
 -R ${DNA_REF}\
 -L ${INTERVAL_LIST}\
 -O ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_CNN.vcf

bgzip ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_CNN.vcf
tabix ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_CNN.vcf.gz

##Applying tranche filtering

gatk --java-options -Xmx6g FilterVariantTranches \
 -V ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_CNN.vcf.gz\
 -L ${INTERVAL_LIST}\
 --resource ${DBSNP}\
 --info-key CNN_1D\
 -O ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_filtered_trenches.vcf
```

```
bgzip ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_filtered_trenches.vcf
tabix ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_filtered_trenches.vcf.gz

##Adding gene names and indexing VCf VCF file

bcftools annotate\
 --threads ${THREADS}\
 --annotations ${INTERVAL_LIST}\
 -h ${HEADER}\
 -c CHROM,FROM,TO,Gene\
 --output-type z\
 --output ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_with_names.vcf.gz\
 ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_filtered_trenches.vcf.gz

tabix ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_with_names.vcf.gz

##Annotating VCF file

bcftools annotate\
 --threads ${THREADS}\
 -a ${DBSNP}\
 -c ID,INFO/RS,INFO/COMMON\
 --output-type z\
 --output ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_dbSNP.vcf.gz\
 ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_with_names.vcf.gz

tabix ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_dbSNP.vcf.gz

##Further filtering of variants

bcftools view\
 --threads ${THREADS}\
 -i 'INFO/RS!="." & FILTER="PASS" & FMT/DP > 9'\
 --regions-file ${IMPRINTED_INTERVAL_LIST}\
 --output-type z\
 --output-file ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_dna.vcf.gz\
 ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_dbSNP.vcf.gz

tabix ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_dna.vcf.gz

##Creating ASE output

gatk ASEReadCounter\
    -R ${REF_GENOME}\
    -I ${PROJECT_DIR}/bam/${SAMPLE}_recalibrated.bam\
    -V ${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_dna.vcf.gz\
    -O ${PROJECT_DIR}/DNA_VCF /${DNA_SAMPLE}_ASE

##Creating DNA variant table
bcftools query\
 -H -f
'%CHROM\t%POS\t%ID\t%INFO/Gene\t%INFO/COMMON\t%REF\t%ALT[\t%GT:%AD{0}:%AD
{1}:%DP]\n'\
 --output ${PROJECT_DIR}/DNA_VCF/dna_final_table.txt\
```

```
${PROJECT_DIR}/DNA_VCF/${DNA_SAMPLE}_dna.vcf.gz
```

Supplementary File 1-Main script

The main UNIX script which includes all set up steps and steps 1-19,29, 40-59 of the pipeline.

SUPPLEMENTARY FILE 2 - XENOFILTER

```
#!/usr/bin/env Rscript
library("XenofilteR")
args <- commandArgs(trailingOnly = TRUE)
sample.list <- data.frame(human = args[1], mouse=args[2])
bp.param <- SnowParam(workers = 8, type = "SOCK")
XenofilteR(sample.list, destination.folder = args[3], bp.param =
bp.param)
```

Supplementary File 2 – XenoFilteR

R script for filtering mouse reads, step 8 of the pipeline.

SUPPLEMENTARY FILE 3 - XCI.R

```
library(stringr)
library(dplyr)

setwd("C:/user/count_and_variant_table_chrX")

##loading the count table

sample_name <- "ERR3466738"
dataset <- read.table(paste0(sample_name,"_edited_counts.txt"),
header=TRUE, sep="\t")
colnames(dataset) <- c('Gene.ID', 'Chr', 'Start', 'Length', 'gene_name',
sample_name)
dataset$Chr <- str_replace(dataset$Chr, ';.*', '')
dataset$Start <- str_replace(dataset$Start, ';.*', '')
colnames(dataset) <- str_replace(colnames(dataset), '.*/', '')
colnames(dataset) <- str_replace(colnames(dataset), '_edited.*', '')

##Normalizing gene expression-TPM

counts <- dataset[-c(1:5)]
rpk= counts / dataset$Length #first normalize for gene length
tot_RPK_per_samp=apply(rpk,2,sum)/10^6 #Sum the normalized values
tpm=t(t(rpk)/tot_RPK_per_samp) # Normalize to library size
tpm[tpm < 1] <- 1
tpm <- cbind(dataset[1:5], tpm)
##Calculating expresion of Y-linked genes

X_degenerate=c("SRY","RPS4Y1","ZFY","AMELY","TBL1Y","PRKY","USP9Y","DBY",
"UTY",

"TMSB4Y","NLGN4Y","CYorf15A","CYorf15B","SMCY","EIF1AY","RPS4Y2")

tpm_y = tpm[tpm$gene_name %in% X_degenerate,]

##Counting expressed genes

per_chrm_n_expressed <- as.data.frame(tpm[-c(1,3,4,5)] %>%
                                        group_by(Chr) %>%
                                        summarise_all(., function(x)
{ sum(x>1)}))

row.names(per_chrm_n_expressed) <- per_chrm_n_expressed$Chr
per_chrm_n_expressed <- per_chrm_n_expressed[-1]

##Loading variant table

variants.file <- read.delim("xci_final_table.txt", stringsAsFactors = F)

names(variants.file) <- str_replace(names(variants.file), 'X\\.+[0-
9]+\\.', '')
names(variants.file) <- str_replace(names(variants.file), '\\.GT.*',
':genotype;allele1;allele2;sum')
```

```
variants.file <- variants.file[(grepl('[A-Z]+',variants.file$Gene)) &
                                 (grepl('^[ACGT]{,1}$',
variants.file$REF)) &
                                 (grepl('^[ACGT]{,1}$',
variants.file$ALT)), -c(3,5,6,7)]
##Removing X-escapees from analysis

X_esacpe=read.csv("X_gene_escaping.csv") #includes escaping and variable

variants.file <- variants.file[!variants.file$Gene %in%
X_esacpe$Gene.name,]

##Computing bialelic ratio

return_allelic_freq <- function(Sample) {

  snp_info <- data.frame(str_split(Sample, ':', simplify = T))
  snp_info[, c(2:4)] <- apply(snp_info[, c(2:4)], 2, as.numeric)

  allelic_freq <- apply(snp_info[,c(2:3)], 1, function(x){return(min(x) /
max(x))})
  snp_info <- cbind(snp_info, allelic_freq)

  heterozygot_pos <- grep('^0/1$', snp_info[,1])
  homozigot_pos <- grep('^0/1$', snp_info[,1], invert = T)
  snp_info[homozigot_pos, 5] <- 0

  snp_info[snp_info[,5] < 0.2, 5] <- 0

  Sample <- snp_info[, 5]
  Sample <- as.numeric(Sample)
  Sample[Sample == 0] <- NaN

  return(Sample)
}

allele_table <- cbind.data.frame(variants.file[1], apply(variants.file[-
c(1:3)], 2, return_allelic_freq))
colnames(allele_table) <- str_replace(colnames(allele_table), ':.*', '')

##Normalizing allelic ratio chrX vs. autosomes

allele_table <- as.data.frame(allele_table %>%
                                group_by(CHROM) %>%
                                summarise_all(., function(x)
{sum(!is.na(x))}))

allele_table <- allele_table[grepl('^chr[0-9]{1,2}$|chrX',
allele_table$CHROM),]

row.names(allele_table) <- allele_table$CHROM

allele_table <- allele_table[-1]
```

```
per_chrm_n_expressed <- per_chrm_n_expressed[row.names(allele_table),
colnames(allele_table), drop = FALSE]

normalised_snp_sum <- allele_table / per_chrm_n_expressed

x_a_ratios <- apply(normalised_snp_sum, 2, function(x) {
  x[row.names(normalised_snp_sum) == 'chrX'] /
mean(x[row.names(normalised_snp_sum) != 'chrX'])
})

#reordering y_tpm and calculating mean y expression

tpm_y <- tpm_y[-c(1:5)][colnames(allele_table)]

mean_Y = apply(tpm_y, 2, mean, na.rm = T)

##Creating final output table
scores <- data.frame(list(x_vs_a_allelic_ratios = x_a_ratios,
Y_av_exp=mean_Y))

output_table <-
  data.frame(X_status = apply(scores, 1, function(x) {
    if(x['x_vs_a_allelic_ratios'] > 0.25 ) return('XaXa')
    else if(x['x_vs_a_allelic_ratios'] > 0.08) return('XaXe')
    else if(x['x_vs_a_allelic_ratios'] < 0.08 & x['Y_av_exp'] < 5)
return('XaXi')
    else if(x['x_vs_a_allelic_ratios'] < 0.08 & x['Y_av_exp'] > 5)
return('Male')
  }))
#Biallelic genes and location
allele_table <- cbind.data.frame(variants.file[c(1,3)],
apply(variants.file[-c(1:3)], 2, return_allelic_freq))
colnames(allele_table) <- str_replace(colnames(allele_table), ':.*', '')

allele_table <- as.data.frame(allele_table %>%
                                group_by(CHROM,Gene) %>%
                                summarise_all(.,function(x)
{sum(!is.na(x))}))

allele_table <- allele_table%>%filter(CHROM=="chrX")

row.names(allele_table) <- allele_table$Gene

allele_table <- allele_table[-c(1:2)]

tpm <- tpm[tpm$gene_name %in% row.names(allele_table),]

tpm <- tpm[!duplicated(tpm$gene_name),]

row.names(tpm) <- tpm$gene_name

tpm <- tpm[row.names(allele_table),]
tpm <- tpm[c('Start', colnames(allele_table))]
```

```
allele_table[] <- apply(allele_table, 2, function(x) {paste0('Biallelic
by ', x, ' SNPs')})
#keeping allele scores only for expressed genes
allele_table[tpm[-1] == 1] <- 'Unexpressed'

#changing biallelic score matrix nan values with 0
allele_table[allele_table == 'Biallelic by 0 SNPs'] <- 'Uninformative'

allele_table <- merge(tpm[1], allele_table, by = 'row.names')

row.names(allele_table) <- allele_table$Row.names

allele_table <- allele_table[-1]

allele_table <- allele_table[order(as.numeric(allele_table$Start)),]



if (output_table[sample_name,] == 'Male') {
  allele_table[sample_name] = 'Male'
}

output_table_genes <- allele_table

write.csv(output_table, "XCI_status.csv")
write.csv(output_table_genes,"biallelic X-linked genes.csv")
```

Supplementary File 3 – XCI

R script for analysis of X chromosome status. Includes steps 20-30 of the pipeline.

SUPPLEMENTARY FILE 4 - LOI WITHOUR DNA INTEGRATION.R

```
library(stringr)
library(dplyr)

setwd("C:/user/count_and_variant_table_LOI")

sample_name <- "ERR3466738"


dataset <- read.table(paste0(sample_name,"_edited_counts.txt"),
header=TRUE, sep="\t")
dataset <- dataset[-c(2:3)]
colnames(dataset) <- c('Gene.ID', 'Length', 'gene_name', sample_name)
colnames(dataset) <- str_replace(colnames(dataset), '.*/', '')


counts <- dataset[-c(1:3)]
rpk= counts / dataset$Length #first normalize for gene length
tot_RPK_per_samp=apply(rpk,2,sum)/10^6 #Sum the normalized values
tpm=t(t(rpk)/tot_RPK_per_samp) # Normalize to library size
tpm <- cbind(dataset[1:3], tpm)


variants.file <- read.delim('loi_final_table.txt', stringsAsFactors = F)
gene_locations <- read.csv('./Imprinted_genes_location.csv')


names(variants.file) <- str_replace(names(variants.file), 'X\\.+[0-
9]+\\.', '')
names(variants.file) <- str_replace(names(variants.file), '\\.GT.*',
':genotype;allele1;allele2;sum')

variants.file <- variants.file[(grepl('[A-Z]+',variants.file$Gene)) &
                                 (grepl('^[ACGT]{,1}$',
variants.file$REF)) &
                                 (grepl('^[ACGT]{,1}$',
variants.file$ALT)), -c(3,5,6,7)]

rownames(variants.file) <- NULL

variants.file <- left_join(gene_locations, variants.file, by = 'Gene',
multiple = 'all')

variants.file[is.na(variants.file)] <- './.:.:.:.'
variants.file <- variants.file[-c(4:5)]


return_allelic_freq <- function(Sample) {

  has_info <- grep('\\./\\.:\\.:\\.:\\.', Sample, invert = TRUE)
  no_info <- grep('\\./\\.:\\.:\\.:\\.', Sample)
  Sample[no_info] <- NaN
```

```
  snp_info <- data.frame(str_split(Sample[has_info], ':', simplify = T))
  snp_info[, c(2:4)] <- apply(snp_info[, c(2:4)], 2, as.numeric)

  allelic_freq <- apply(snp_info[,c(2:3)], 1, function(x){return(min(x) /
max(x))})
  snp_info <- cbind(snp_info, allelic_freq)

  heterozygot_pos <- grep('^0/1$', snp_info[,1])
  homozigot_pos <- grep('^0/1$', snp_info[,1], invert = T)
  snp_info[homozigot_pos, 5] <- 0

  snp_info[snp_info[,5] < 0.2, 5] <- 0
  snp_info[snp_info[, 4] < 10, 5] <- 0

  Sample[has_info] <- snp_info[, 5]
  Sample <- as.numeric(Sample)
  Sample[Sample == 0] <- NaN

  return(Sample)
}

allele_table <- cbind.data.frame(variants.file[c(1:3)],
apply(variants.file[-c(1:3)], 2, return_allelic_freq))
colnames(allele_table) <- str_replace(colnames(allele_table), ':.*', '')


allele_table <- as.data.frame(allele_table %>%
  group_by(Gene = factor(Gene, levels = unique(Gene)),
Location,Imprinting.related.disease) %>%
  summarise_all(.,function(x) {sum(!is.na(x))}))

row.names(allele_table) <- allele_table$Gene
allele_table <- allele_table[,-1]


tpm <- tpm[tpm$gene_name %in% row.names(allele_table),]
row.names(tpm) <- tpm$gene_name
tpm <- tpm[row.names(allele_table),]
tpm <- tpm[colnames(allele_table)[-c(1:2)]]

allele_table[-c(1:2)] <- apply(allele_table[-c(1:2)],2,function(x)
{paste0('Biallelic by ', x, ' SNPs')})
allele_table[-c(1:2)][tpm < 1] <- 'Unexpressed'
allele_table[-c(1:2)][allele_table[-c(1:2)] == 'Biallelic by 0 SNPs'] <-
'Uninformative'


#Per gene LOI
write.csv(allele_table, file = './LOI_matrix.csv', row.names = T)

#Per region LOI
allele_table <- as.data.frame(allele_table[-2] %>%
```

```
                                group_by(Location = factor(Location,
levels = unique(Location))) %>%
                                summarise_all(., function(x) {sum(x !=
'Uninformative' & x != 'Unexpressed')}))
row.names(allele_table) <- allele_table$Location
allele_table <- allele_table[-1]
allele_table[] <- apply(allele_table, 2, function(x) {paste0(x, ' genes
with biallelic expression')})
write.csv(allele_table, file = './LOI_per_locus_matrix.csv', row.names =
T)
```

Supplementary File 4 – LOI analysis without DNA integration

R script for analyzing imprinting status, without DNA integration. Includes steps 32-39 of the pipeline.

SUPPLEMENTARY FILE 5 - LOI WITH DNA INTEGRATION.R

```
library(stringr)
library(dplyr)

setwd("C:/user/count_and_variant_tables_LOI")

sample_name='ERR3466738'
dna_sample_name='SRR2070629'
gene_locations <- read.csv('./Imprinted_genes_location.csv')
dna.file <- read.delim('./dna_final_table.txt', stringsAsFactors = F)
variants.file <- read.delim(paste0(dna_sample_name,'_ASE'),
stringsAsFactors = F)

names(dna.file) <- str_replace(names(dna.file), 'X\\.+[0-9]+\\.', '')
names(dna.file) <- str_replace(names(dna.file), '\\.GT.*',
':genotype;allele1;allele2;sum')
dna.file <- dna.file[(grepl('[A-Z]+',dna.file$Gene)) &
                               (grepl('^[ACGT]{,1}$', dna.file$REF)) &
                               (grepl('^[ACGT]{,1}$', dna.file$ALT)), -
5]
rownames(dna.file) <- NULL

heterozygote_pos <- inner_join(gene_locations, dna.file, by = 'Gene',
multiple = 'all')
variants.file <- variants.file[1:8]
colnames(variants.file)[1:5] <- c('CHROM','POS','ID','REF','ALT')
#process dataframe and keep variants that are annotated to a gene's exon
(Gene),
#that have a dbSNP ID (RS) and are one base-pair long
merged_table = inner_join(heterozygote_pos, variants.file, multiple =
'all')
return_allelic_freq <- function(snp) {
  if(as.numeric(snp['totalCount']) < 10) return(NaN)
  else {
    return(min(as.numeric(snp[c('refCount', 'altCount')])) /
max(as.numeric(snp[c('refCount', 'altCount')])))
  }
}

merged_table$freq <- apply(merged_table, 1, return_allelic_freq)

merged_table$allelic <- sapply(merged_table$freq, function(x) {
  if(is.na(x)) return(0)
  else if (x < 0.1) return(1)
  else if (x > 0.2) return(2)
  else return(0)
})

merged_table <- as.data.frame(merged_table %>%
  group_by(Gene = factor(Gene, levels = unique(Gene)),
Location,Imprinting.related.disease) %>%
```

```
  summarise(n.biallelic = sum(allelic == 2), n.monoallelic = sum(allelic
== 1), n.uninformative = sum(allelic == 0)))


merged_table$Sample <- apply(merged_table[-c(1:3)],1, function(x) {
  if(x['n.monoallelic'] == 0 & x['n.biallelic'] == 0)
{return('Uninformative')}
  else if (x['n.biallelic'] == 0) {return(paste0('Monoallelic by ',
x['n.monoallelic'], ' SNPs'))}
  else {return(paste0('Biallelic by ', x['n.biallelic'], ' SNPs'))}
})
colnames(merged_table)[7] <- sample_name
merged_table <- merged_table[c(1,2,3,7)]
row.names(merged_table) <- merged_table$Gene
merged_table <- merged_table[-1]


dataset <- read.table(paste0(sample_name, "_edited_counts.txt"),
header=TRUE, sep="\t")
dataset <- dataset[-c(2:3)]
colnames(dataset) <- c('Gene.ID',  'Length', 'gene_name', sample_name)
colnames(dataset) <- str_replace(colnames(dataset), '.*/', '')
colnames(dataset) <- str_replace(colnames(dataset), '_edited.*', '')


counts <- dataset[-c(1:3)]
rpk= counts / dataset$Length #first normalize for gene length
tot_RPK_per_samp=apply(rpk,2,sum)/10^6 #Sum the normalized values
tpm=t(t(rpk)/tot_RPK_per_samp) # Normalize to library size
tpm <- cbind(dataset[1:3], tpm)


tpm <- tpm[tpm$gene_name %in% row.names(merged_table),]
row.names(tpm) <- tpm$gene_name
tpm <- tpm[row.names(merged_table),]
tpm <- tpm[colnames(merged_table)[-c(1:2)]]


merged_table[-c(1:2)][tpm < 1] <- 'Unexpressed'
merged_table <- merged_table[apply(merged_table[-c(1:2)], 1, function(x)
{sum(x == 'Unexpressed') + sum(x == 'Uninformative') != length(x)}),]
output_table_per_gene <- merged_table
write.csv(output_table_per_gene, file =
'./LOI_matrix_with_dna_integration_new.csv', row.names = T)
```

Supplementary file 5 – LOI analysis with DNA integration

R script for analyzing imprinting status, with DNA integration. Includes steps 60-67 of the pipeline.