**Supplemental Data S8 Essential computing codes used in this study.**

The full description of the codes and related data files are available at Gitlab: <https://gitlab.gwdg.de/wenyu.zhang/mouse_population_isoform/>.

##############################################################################

###### **Part I: selection of cDNA Library Enrichment protocol for PacBio Iso-Seq**

## Three types of cDNA library enrichment methods. The former protocol selectively synthesizes cDNA molecules from transcripts with a poly-A tail, while the latter two protocols synthesize cDNA molecules from transcripts with both a 5’ cap and a poly-A tail.

# i) Standard PacBio SMARTer PCR cDNA Synthesis kit (Clontech Laboratories, Inc.

# ii) TeloPrime Full-Length cDNA Amplification Kit V1 (Lexogen GmbH), combined with oligo-dT primer from the SMARTer PCR cDNA Synthesis kit

# iii) TeloPrime Full-Length cDNA Amplification Kit V2 (Lexogen GmbH), combined with oligo-dT primer from the SMARTer PCR cDNA Synthesis kit

####### Data analysis steps:

## IsoSeq3 v3.4.0

## ccs v6.0.0

## lima v2.0.0

## minimap2 v2.24-r1122

## samtools v1.9

## k8-Linux v0.2.5-r80

## UCSC kentUtils (bedToGenePred, genePredToGtf): http://hgdownload.soe.ucsc.edu/admin/exe/

## gffcompare v0.12.2

## GRCm39/mm39 reference genome data: http://ftp.ensembl.org/pub/release-103/fasta/mus\_musculus/dna/

## Ensembl v103 gtf annotation data: http://ftp.ensembl.org/pub/release-103/gtf/mus\_musculus/

# 1) Generate circular consensus sequences (CCS) from sub-read data of each SMRT cell

$ ccs --min-passes 3 --min-length 50 --max-length 1000000 --min-rq 0.99 --reportFile /PathTo/Clontech\_Polished.ccs.report /PathTo/Clontech.subreads.bam /PathTo/Clontech\_Polished.ccs.bam

$ ccs --min-passes 3 --min-length 50 --max-length 1000000 --min-rq 0.99 --reportFile /PathTo/TeloPrimeV1\_Polished.ccs.report /PathTo/TeloPrimeV1.subreads.bam /PathTo/TeloPrimeV1\_Polished.ccs.bam

$ ccs --min-passes 3 --min-length 50 --max-length 1000000 --min-rq 0.99 --reportFile /PathTo/TeloPrimeV2\_Polished.ccs.report /PathTo/TeloPrimeV2.subreads.bam /PathTo/TeloPrimeV2\_Polished.ccs.bam

# 2) Remove cDNA primers and orientate the CCS reads

$ lima --isoseq /PathTo/Clontech\_Polished.ccs.bam /PathTo/Clontech\_Primer.fasta /PathTo/Clontech\_Polished.ccs.demux.bam

$ lima --isoseq /PathTo/TeloPrimeV1\_Polished.ccs.bam /PathTo/Lexogen\_Primer.fasta /PathTo/TeloPrimeV1\_Polished.ccs.demux.bam

$ lima --isoseq /PathTo/TeloPrimeV2\_Polished.ccs.bam /PathTo/Lexogen\_Primer.fasta /PathTo/TeloPrimeV2\_Polished.ccs.demux.bam

# 3) Tag and remove the 10-mer unique molecular identifier (UMI) from the above demuxed CCS reads.

$ isoseq3 tag --design T-10U /PathTo/Clontech\_Polished.ccs.demux.primer\_5p--primer\_3p.bam /PathTo/Clontech.flt.bam

$ isoseq3 tag --design T-10U /PathTo/TeloPrimeV1\_Polished.ccs.demux.primer\_5p--primer\_3p.bam /PathTo/TeloPrimeV1.flt.bam

$ isoseq3 tag --design T-10U /PathTo/TeloPrimeV2\_Polished.ccs.demux.primer\_5p--primer\_3p.bam /PathTo/TeloPrimeV2.flt.bam

# 4) Remove reads with concatemers and find reads with polyA (>=20bp) and remove it

$ isoseq3 refine --require-polya /PathTo/Clontech.flt.bam /PathTo/Clontech\_Primer.fasta /PathTo/Clontech.fltnc.bam

$ isoseq3 refine --require-polya /PathTo/TeloPrimeV1.flt.bam /PathTo/Lexogen\_Primer.fasta /PathTo/TeloPrimeV1.fltnc.bam

$ isoseq3 refine --require-polya /PathTo/TeloPrimeV2.flt.bam /PathTo/Lexogen\_Primer.fasta /PathTo/TeloPrimeV2.fltnc.bam

# 5) PCR deduplication of fltnc reads based on the UMI tag information （remove redundance from the same founder molecular）

$ isoseq3 dedup /PathTo/Clontech.fltnc.bam /PathTo/Clontech.fltnc.dedup.bam

$ isoseq3 dedup /PathTo/TeloPrimeV1.fltnc.bam /PathTo/TeloPrimeV1.fltnc.dedup.bam

$ isoseq3 dedup /PathTo/TeloPrimeV2.fltnc.bam /PathTo/TeloPrimeV2.fltnc.dedup.bam

# 6) De novo clustering based on the similarity of the reads（remove redundance from the same type of transcript): https://github.com/PacificBiosciences/IsoSeq

$ isoseq3 cluster /PathTo/Clontech.fltnc.dedup.bam /PathTo/Clontech.fltnc.dedup.clustered.bam --verbose --use-qvs --singletons

$ isoseq3 cluster /PathTo/TeloPrimeV1.fltnc.dedup.bam /PathTo/TeloPrimeV1.fltnc.dedup.clustered.bam --verbose --use-qvs --singletons

$ isoseq3 cluster /PathTo/TeloPrimeV2.fltnc.dedup.bam /PathTo/TeloPrimeV2.fltnc.dedup.clustered.bam --verbose --use-qvs --singletons

# 7) Align all the high-confidence isoforms (with >= 2 supporting FLNC reads) to the mm39 reference genome with minimap2

# Generate the index file for genome assembly data

$ minimap2 -d /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly.fa.mmi /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly.fa

# Do the alignment to mm39 reference genome with minimap2

$ minimap2 -ax splice:hq -uf --secondary=no -C5 -O6,24 -B4 /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly.fa.mmi /PathTo/Clontech.clustered.isoform.fa | samtools view -hS - | samtools sort -O SAM ->/PathTo/Clontech.sorted.sam

$ minimap2 -ax splice:hq -uf --secondary=no -C5 -O6,24 -B4 /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly.fa.mmi /PathTo/TeloPrimeV1.clustered.isoform.fa | samtools view -hS - | samtools sort -O SAM ->/PathTo/TeloPrimeV1.sorted.sam

$ minimap2 -ax splice:hq -uf --secondary=no -C5 -O6,24 -B4 /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly.fa.mmi /PathTo/TeloPrimeV2.clustered.isoform.fa | samtools view -hS - | samtools sort -O SAM ->/PathTo/TeloPrimeV2.sorted.sam

# 8) Convert the above spliced alignment sam file into gtf annotation format

# https://github.com/lh3/minimap2/blob/master/misc/README.md

# http://onetipperday.sterding.com/2012/08/convert-bed-to-gtf.html

$ k8-Linux /PathTo/minimap2/misc/paftools.js splice2bed -m /PathTo/Clontech.sorted.sam stdout | bedToGenePred stdin stdout | genePredToGtf file stdin /PathTo/Clontech.gtf

$ k8-Linux /PathTo/minimap2/misc/paftools.js splice2bed -m /PathTo/TeloPrimeV1.sorted.sam stdout | bedToGenePred stdin stdout | genePredToGtf file stdin /PathTo/TeloPrimeV1.gtf

$ k8-Linux /PathTo/minimap2/misc/paftools.js splice2bed -m /PathTo/TeloPrimeV2.sorted.sam stdout | bedToGenePred stdin stdout | genePredToGtf file stdin /PathTo/TeloPrimeV2.gtf

# 9) Compare the above gtf file to the Ensembl v103 gtf annotations with gffcompare

$ gffcompare -T -V -r /PathTo/Mus\_musculus.GRCm39.103.gtf /PathTo/Clontech.gtf -o /PathTo/Clontech

$ gffcompare -T -V -r /PathTo/Mus\_musculus.GRCm39.103.gtf /PathTo/TeloPrimeV1.gtf -o /PathTo/TeloPrimeV1

$ gffcompare -T -V -r /PathTo/Mus\_musculus.GRCm39.103.gtf /PathTo/TeloPrimeV2.gtf -o /PathTo/TeloPrimeV2

##############################################################################

###### **Part II: selection of cDNA Library Enrichment protocol for PacBio Iso-Seq**

###### This part shows the steps and codes for PacBio Iso-Seq long read data, mainly following IsoSeq3 pipeline: https://github.com/PacificBiosciences/IsoSeq.

####### Data analysis steps:

## IsoSeq3 v3.4.0

## ccs v6.0.0

## lima v2.0.0

## minimap2 v2.24-r1122

## samtools v1.9

## TAMA program

## Kallisto v0.46.2

## SQANTI3 v4.2

## SUPP2

# 1) Generate circular consensus sequences (CCS) from sub-read data of each SMRT cell

$ ccs --min-passes 3 --min-length 50 --max-length 1000000 --min-rq 0.99 --reportFile /PathTo/SMRTCell\_Polished.ccs.report /PathTo/SMRTCell.subreads.bam /PathTo/SMRTCell\_Polished.ccs.bam

# 2) Merge the ccs data of 3 SMRT cells from the same sample into one and store into /PathTo/SampleID.ccs.fofn

# For instance: GE\_1.fltnc.fofn

/PathTo/m54062\_190406\_032202\_Polished.ccs.bam

/PathTo/m54144\_190401\_144814\_Polished.ccs.bam

/PathTo/m54124\_200314\_061109\_Polished.ccs.bam

# 3) Remove cDNA primers and orientate the CCS reads

$ lima --isoseq /PathTo/SampleID\_Polished.ccs.bam /PathTo/Lexogen\_Primer.fasta /PathTo/SampleID\_Polished.ccs.demux.bam

# 4) Tag and remove the 10-mer unique molecular identifier (UMI) from the above demuxed CCS reads.

$ isoseq3 tag --design T-10U /PathTo/SampleID\_Polished.ccs.demux.primer\_5p--primer\_3p.bam /PathTo/SampleID.flt.bam

# 5) Remove reads with concatemers and find reads with polyA (>=20bp) and remove it

$ isoseq3 refine --require-polya /PathTo/SampleID.flt.bam /PathTo/Lexogen\_Primer.fasta /PathTo/SampleID.fltnc.bam

# 6) PCR deduplication of fltnc reads based on the UMI tag information (remove redundance from the same founder molecular)

$ isoseq3 dedup /PathTo/SampleID.fltnc.bam /PathTo/SampleID.fltnc.dedup.bam

# 7) De novo clustering based on the similarity of the reads（remove redundance from the same type of transcript): https://github.com/PacificBiosciences/IsoSeq

$ isoseq3 cluster /PathTo/SampleID.fltnc.dedup.bam /PathTo/SampleID.fltnc.dedup.clustered.bam --verbose --use-qvs --singletons

# 8) Align all the high-confidence isoforms (with >= 2 supporting FLNC reads) to the mm39 reference genome with minimap2

# Generate the index file for genome assembly data

$ minimap2 -d /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly.fa.mmi /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly.fa

# Do the alignment to mm39 reference genome with minimap2

$ minimap2 -ax splice:hq -uf --secondary=no -C5 -O6,24 -B4 /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly.fa.mmi /PathTo/SampleID.clustered.isoform.fa | samtools view -hS - | samtools sort -O SAM ->/PathTo/SampleID.sorted.sam

# 9) Collapse and merge similar isoforms from all 48 samples into a single high-quality transcriptome

#TAMA collapse: collape the similar isoforms for each splited sam file based on genomic coordiate of minimap2 alignment

$ tama\_collapse.py -s /PathTo/SampleID.sorted.sam -f /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly.fa -p ./PathTo/$Sample\_ID -d merge\_dup -x capped -c 95 -i 90 -icm ident\_map -m 5 -a 1000 -z 30 -sj sj\_priority

#TAMA merge: merge the similar isoforms from all 48 samples into a single high-quality transcriptome

$ tama\_merge.py -f /PathTo/All\_Sample\_Bed.filelist -p /PathTo/All\_Sample -d merge\_dup -m 5 -a 1000 -z 30

# 10) Quantification of the isoform/transcript expression

#Method one: on the basis of the supporting FLNC reads for each isoform in each sample (output from SQUANTI3)

#Method two: on the basis of the alignment of Illumina RNA-Seq data to the merged isoforms (using Kallisto)

$ kallisto index -i /PathTo/All\_Sample\_Merged\_Isoform.fa.idx /PathTo/All\_Sample\_Merged\_Isoform.fa

$ kallisto quant -i /PathTo/All\_Sample\_Merged\_Isoform.fa.idx -o /PathTo/Sample\_RF --bias --fusion --rf-stranded /PathTo/SampleID\_filtered\_R1.fastq /PathTo/SampleID\_filtered\_R2.fastq

# 11) Quality control of the merged transcriptome with SQUANTI3

#Long read-defined transcriptome: TAMA merged transcripts for all sample

#Reference annotation gtf: Ensembl v103

#Reference genome seq: GRCm39/mm39

#Cage-peak(TSS) data: Provided by SQANTI3 (ref\_TSS)

#PolyA site data: https://polyasite.unibas.ch/download/atlas/2.0/GRCm38.96/atlas.clusters.2.0.GRCm38.96.bed.gz

#PolyA motif data: From PolyASite database for mouse:https://polyasite.unibas.ch/atlas#2

#FL count information: FL count for each merged transcript (based on FLTNC reads)

#STAR splice junction data: STAR alignment outputs from all 48 samples

#Short read BAMs: The path to STAR alignment bam files from all 48 samples

$ python /PathTo/sqanti3\_qc.py /PathTo/All\_Sample\_Merged.gtf /PathTo/Mus\_musculus.GRCm39.103.gtf /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly.fa --cage\_peak /PathTo/mouse.refTSS\_v3.1.bed --polyA\_peak /PathTo/Mm39\_PolyASite.bed --polyA\_motif\_list /PathTo/PolyASite\_mouse\_polaA\_motif.txt -fl /PathTo/Merged\_Isoform\_Abundance.txt -c /PathTo/STAR\_Alignment\_Bam/ --SR\_bam /PathTo/STAR\_Alignment\_Bam/ -o /PathTo/All\_Sample -d /PathTo/SQANTI3\_QC/ --report both

# 12) Filter out low-quality isoforms from the merged transcriptome

#Filters applied:

#(a) min # of exons: 2; remove all mono-exon isoforms. exons >=2

#(b) Reliable 3'end: perc\_A\_downstream\_TTS <= 60%

#(c) For Junction site: No junction should be labeled as RTSwitching && (all splice sites should be canonical || supported by at least three spanning reads based on STAR junction output file)

#(d) For non-FSM transcript, then at least one of the three criteria should be met: i) within 50bp of TSSs in the reference annotation; ii) Within the cage peak defined by SQUANTI3; iii) Ratio TSS >=1.5

# 13) Detection of local AS events with SUPP2

$ suppa.py generateEvents -i /PathTo/All\_Sample\_Merged.gtf -o /PathTo/All\_Isoform -f ioe -e SE SS MX RI FL

##############################################################################

###### **Part III: Illumina RNA-Seq data analysis**

## The first section shows the steps and codes to generate gene expression levels with Illumina RNA-Seq data

####### Data analysis steps:

## fastp v0.20.0

## STAR v2.7.0e

## featureCounts v1.6.3

## R v4.1.0

## R package ggfortify v0.4.8

## GRCm39/mm39 reference genome data: http://ftp.ensembl.org/pub/release-103/fasta/mus\_musculus/dna/

## Ensembl v103 gtf annotation data: http://ftp.ensembl.org/pub/release-103/gtf/mus\_musculus/

# 1) Trim the raw fastq data with fastp. Only keep paired-end reads (both ends) >= 50bp and average quality score >=20 for further analysis.

$ fastp -i /PathTo/SampleID\_R1.fastq.gz -o /PathTo/SampleID\_Trimmed\_R1.fastq -I /PathTo/SampleID\_R2.fastq.gz -O /PathTo/SampleID\_Trimmed\_R2.fastq --cut\_front --average\_qual 20 --length\_required 50 -j /PathTo/SampleID\_Trimming.json -h /PathTo/SampleID\_Trimming.html

# 2) Map the trimmed fastq data to GRCm39/mm39 reference genome with STAR. With optimised paramter setting to compensate the sequence divergences of individuals from various populations and species.

$ STAR --runMode genomeGenerate --genomeDir /PathTo/GRCm39\_genomeDir\_STAR --genomeFastaFiles /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly.fa --sjdbGTFfile /PathTo/Mus\_musculus.GRCm39.103.gtf --sjdbOverhang 149

$ STAR --runMode alignReads --twopassMode Basic --twopass1readsN -1 --genomeDir /PathTo/GRCm39\_genomeDir\_STAR --readFilesIn /PathTo/SampleID\_Trimmed\_R1.fastq /PathTo/SampleID\_Trimmed\_R2.fastq --outFileNamePrefix /PathTo/SampleID --outFilterMismatchNmax 30 --scoreDelOpen -1 --scoreDelBase -1 --scoreInsOpen -1 --scoreInsBase -1 --seedSearchStartLmax 25 --winAnchorMultimapNmax 100 --outSAMtype BAM SortedByCoordinate

# 3) Count the fragments uniquely mapped to the annotated genes with featureCounts. In reversely stranded mode, excluding multi-mapping reads and the ones with alignment quality below 5.

$ featureCounts -p -s 2 -Q 5 -t exon -g gene\_id -a /PathTo/Mus\_musculus.GRCm39.103.gtf -o /PathTo/FeatureCounts\_Fragment\_STAR.count /PathTo/Sample\_1\_STAR\_Alignment.bam .... /PathTo/Sample\_N\_STAR\_Alignment.bam

##The second part shows the steps and codes for the analysis of individual relatedness based on SNP variants called from Illumina RNA-Seq dataset. The SNP calling based on RNA-seq data follows GATK best practice version 4: <https://gatk.broadinstitute.org/hc/en-us/articles/360035531192-RNAseq-short-variant-discovery-SNPs-Indels->

####### Data analysis steps:

## GATK v4.1.9

## PICARD: v2.8.0

## samtools v1.9

## vcftools: 0.1.14

## SNP variants called from genomic datasets of same population: http://wwwuser.gwdg.de/~evolbio/evolgen/wildmouse/mouse\_retroCNV/

# 1) Mark and remove the duplicates from STAR alignment bam files with PICARD

$ jar -jar /PathTo/PICARD.jar I=/PathTo/SampleID\_STAR\_Alignment\_Sorted.bam O=/PathTo/SampleID\_STAR\_Alignment\_Sorted\_Nodup.bam REMOVE\_DUPLICATES=true ASSUME\_SORTED=true MAX\_RECORDS\_IN\_RAM=500000 VALIDATION\_STRINGENCY=LENIENT

$ samtools index /PathTo/SampleID\_STAR\_Alignment\_Sorted\_Nodup.bam

# 2) Reorder the bam data with the new order of chr (1,2,3,4...., rather than the original 1,10,11....)

$ java -jar /PathTo/PICARD.jar CreateSequenceDictionary R=/PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly\_Reorder.fa O=/PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly\_Reorder.dict

$ samtools faidx /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly\_Reorder.fa

$ java -Xmx80g -d64 -jar /PathTo/PICARD.jar ReorderSam I=/PathTo/SampleID\_STAR\_Alignment\_Sorted\_Nodup.bam O=/PathTo/SampleID\_STAR\_Alignment\_Sorted\_Nodup\_Reorder.bam R=/PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly\_Reorder.fa

$ samtools index /PathTo/SampleID\_STAR\_Alignment\_Sorted\_Nodup\_Reorder.bam

# 3) Split reads with N in the cigar into multiple supplementary alignments and hard clips mismatching overhangs using the SplitNCigarReads function in GATK

$ gatk SplitNCigarReads -R /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly\_Reorder.fa -I /PathTo/SampleiD\_STAR\_Alignment\_Sorted\_Nodup\_Reorder.bam -O /PathTo/SampleID\_STAR\_Alignment\_Sorted\_Nodup\_Reorder\_splitN.bam

# 4) Recalibrate the alignment bam file based on known SNP data calling from genomic datasets of same population

$ java -jar /PathTo/PICARD.jar AddOrReplaceReadGroups I=/PathTo/SampleID\_STAR\_Alignment\_Sorted\_Nodup\_Reorder\_splitN.bam O=/PathTo/SampleID\_STAR\_Alignment\_Sorted\_Nodup\_Reorder\_splitN\_AddRG.bam RGLB=lib1 RGPL=Illumina RGPU=Sample\_ID RGSM=Sample\_ID ##Add read group info

$ gatk BaseRecalibrator -I /PathTo/SampleID\_STAR\_Alignment\_Sorted\_Nodup\_Reorder\_splitN\_AddRG.bam -R /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly\_Reorder.fa --known-sites /PathTo/Known\_Genomic\_SNP\_Variant.vcf.gz -O /PathTo/SampleID.recal.data.table

$ gatk ApplyBQSR -I /PathTo/SampleID\_STAR\_Alignment\_Sorted\_Nodup\_Reorder\_splitN\_AddRG.bam -R /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly\_Reorder.fa --bqsr-recal-file /PathTo/SampleID.recal.data.table -O /PathTo/SampleID\_STAR\_Alignment\_Sorted\_Nodup\_Reorder\_splitN\_AddRG\_Recalibration.bam

# 5) Call raw genetic variants for each individual using the HaplotypeCaller function in GATK

$ gatk --java-options -Xmx4g HaplotypeCaller -R /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly\_Reorder.fa -I /PathTo/SampleID\_STAR\_Alignment\_Sorted\_Nodup\_Reorder\_splitN\_AddRG\_Recalibration.bam -O ./PathTo/SampleID.g.vcf.gz -ERC GVCF

# 6) Combine all the individual GVCF files into one

$ gatk CombineGVCFs -R /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly\_Reorder.fa --variant /PathTo/Sample\_1.g.vcf.gz --variant /PathTo/Sample\_2.g.vcf.gz ... --variant /PathTo/Sample\_N.g.vcf.gz -O ./Combined\_Sample.g.vcf.gz

# 7) Joint genotyping with gatk GenotypeGVCFs

$ gatk --java-options -Xmx20g GenotypeGVCFs -R /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly\_Reorder.fa -V /PathTo/Combined\_Sample.g.vcf.gz -O /PathTo/Combined\_Sample\_JointGenotyping.g.vcf.gz

# 8) Hard filter the variant file with gatk VariantFiltration

$ gatk VariantFiltration -R /PathTo/Mus\_musculus.GRCm39.dna.primary\_assembly\_Reorder.fa -V /PathTo/Combined\_Sample\_JointGenotyping.g.vcf.gz -O /PathTo/Combined\_Sample\_JointGenotyping\_filtered.vcf.gz --filter-name "my\_snp\_filter" --filter-expression "QD < 2.0 || FS > 60.0 || MQ < 40.0 || MQRankSum < -12.5 || ReadPosRankSum < -8.0 || SOR > 3.0"

# 9) Calculate the relatedness for all possible pairs of individuals using vcftools

$ vcftools --gzvcf /PathTo/Combined\_Sample\_JointGenotyping\_filtered.vcf.gz --remove-indels --remove-filtered-all --chr 1 --chr 1 --chr 2 --chr 3 --chr 4 --chr 5 --chr 6 --chr 7 --chr 8 --chr 9 --chr 10 --chr 11 --chr 12 --chr 13 --chr 14 --chr 15 --chr 16 --chr 17 --chr 18 --chr 19 --chr X --chr Y --max-alleles 2 --min-alleles 2 --max-missing 0.8 --thin 1000000 --relatedness2 --out /PathTo/Pairwise\_Relatedness