

Code availability:

convert cram file to bam; index and sorting bam file:

```
samtools view -b --threads 25 -T GRCh38_full_analysis_set_plus_decoy_hla.fa -o  
${prefix}.bam ${input-file}.cram - this was done for the 1KGP samples.
```

All samples:

```
samtools sort -m 2G --threads 20 -O bam -T tmp -o ${prefix}.bam ${input-file}.bam  
samtools index ${sorted bam file} ${prefix}.bam.bai
```

Remove and mark duplicates:

```
java -Xmx32G -Djava.io.tmpdir=$TMPDIR -jar picard/2.6.0/picard.jar \  
MarkDuplicates INPUT= $sorted.bam file OUTPUT=$outfile.bam \  
METRICS_FILE=rdupl_metrics REMOVE_DUPLICATES=true
```

Qualimap:

```
qualimap bamqc -bam ${input.bam} \  
-nt 40 -outdir qualimap_report --java-mem-size=32G
```

#####

STR calling:

#####

```
HipSTR --bams ${sample.bam} --fasta GRCh38_full_analysis_set_plus_decoy_hla.fa --  
regions ${HipSTR-STR-catalogue-file}.bed \  
--str-vcf ${prefix}.vcf.gz --log ${prefix}.log --viz-out ${prefix}.viz.gz --min-reads 25 --def-  
stutter-model
```

Filtering the raw VCF file:

```
dumpSTR --vcf ${vcffile} --out prefix --vcftype hipstr --hipstr-max-call-DP 1000 --  
hipstr-min-suppl reads 1 --hipstr-max-call-flank-indel 0.15 --hipstr-max-call-stutter  
0.15 -- drop-filtered
```

#####

GangSTR --bam \${file.bam} \

--ref GRCh38_full_analysis_set_plus_decoy_hla.fa --regions \${GangSTR-STR-catalogue-file}.bed --out \${prefix}

Filtering the raw VCF file

dumpSTR --vcf \${vcffile} --out \${prefix} --vcftype gangstr --gangstr-filter-spanbound-only --gangstr-filter-badCI --gangstr-min-call-DP 20 --gangstr-max-call-DP 1000

#####

ExpansionHunter --reads \${file.bam} \

--reference < GRCh38_full_analysis_set_plus_decoy_hla.fa > \

--variant-catalog <JSON file specifying variants to genotype> \

--output-prefix \${prefix}

#####

STRetch/tools/bin/bpipe run -m \${RAM}GB -n \${threads} \

-p input_regions=\${STRetch-STR-catalogue-file}.bed \

STRetch/pipelines/STRetch_wgs_bam_pipeline.groovy \${input-file.bam}

#####

string extract -f \$reference_fasta /path/to/\$sample.cram \$sample.bin

joint calling:

string merge --output-prefix str-results/joint -f \$reference_fasta \$sample1.bin \$sample2.bin

..... \$sampleN.bin

#####

ExpansionHunterDenovo profile --reads \${input-file}.bam \

--reference GRCh38_full_analysis_set_plus_decoy_hla.fa \

--output-prefix \${prefix} --min-anchor-mapq 50 --max-irr-mapq 40 --log-reads

Merge profile

```
ExpansionHunterDenovo merge \  
  
--reference GRCh38_full_analysis_set_plus_decoy_hla.fa --manifest ${manifest-file} --  
  
output-prefix ${prefix}
```

Conduct case-control analysis

```
ExpansionHunterDenovo-v0.9.0-linux_x86_64/scripts/casecontrol.py locus \  
  
--manifest ${manifest-file} \  
  
--multisample-profile ${samples_merged_profile}.json \  
  
--output ${prefix}
```

Conduct outlier analysis

```
ExpansionHunterDenovo-v0.9.0-linux_x86_64/scripts/outlier.py locus \  
  
--manifest ${manifest-file} \  
  
--multisample-profile ${samples_merged_profile}.json \  
  
--output ${prefix}
```

Annotate output file from above using ANNOVAR software

```
bash ExpansionHunterDenovo-v0.9.0-linux_x86_64/scripts/annotate_ehcn.sh \  
  
--ehcn-results ${output-file} \  
  
--ehcn-annotated-results ${prefix} \  
  
--annovar-annotate-variation annovar/annotate_variation.pl \  
  
--annovar-humandb humandb \  
  
--annovar-buildver hg 38
```

#####

Imputation using Beagle

```
java -Xmx4g -jar beagle.r1399.jar gt=input_SNPs.vcf.gz \  
  
ref=${chrom}_final_SNP_merged.vcf.gz \  
  
out=imputed_TR_SNPs.vcf
```

calculate_STR_ld_pegas.R - R code used to calculate LD – based on the methods of Schaid (2004) and Zaykin et al. (2008) for multiallelic markers, implemented in Pegas R package (Paradis, 2010).
