GATK has a ton of steps to call potiental de novo snps
My scripts can be found at `/scratch/Workshop/SR2019/8_GATK/reseqscripts/`

Step 1: Map with the GATK genome and the read groups marked

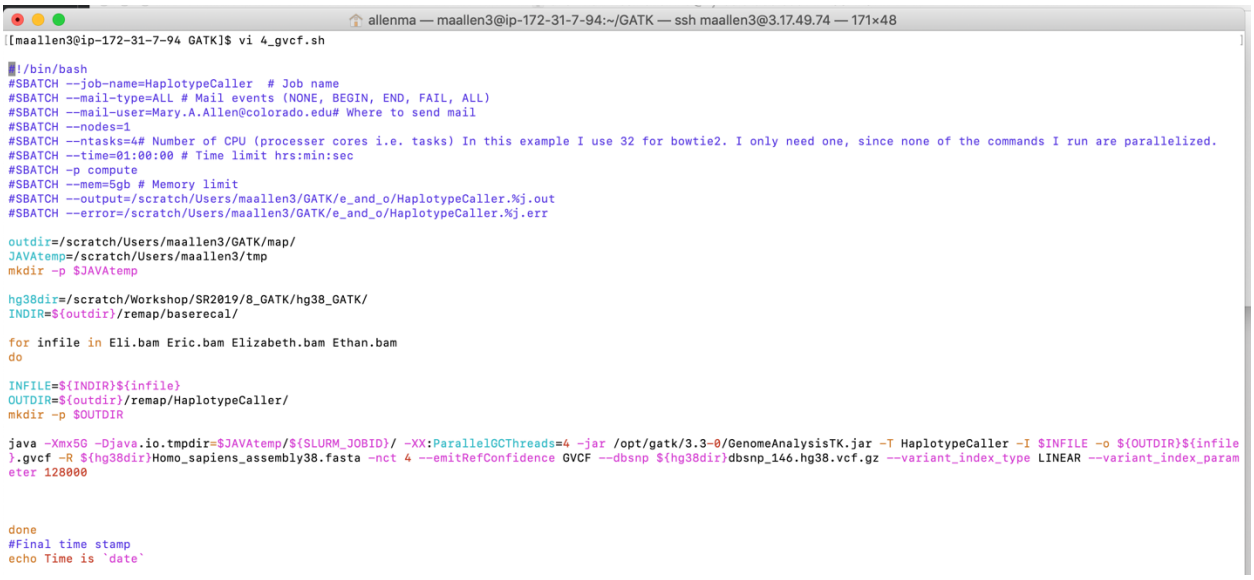
```
#!/bin/bash
#SBATCH --job-name=mapremap  # Job name
#SBATCH --mail-type=ALL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=Mary.A.Allen@colorado.edu# Where to send mail
#SBATCH --nodes=1
#SBATCH --ntasks=4# Number of CPU (processer cores i.e. tasks) In this example I use 32 for bowtie2. I only need one, since no
ne of the commands I run are parallelized.
#SBATCH --time=00:10:00 # Time limit hrs:min:sec
#SBATCH -p compute
#SBATCH --mem=5gb # Memory limit
#SBATCH --output=/scratch/Users/maallen3/GATK/e_and_o/map.%j.out
#SBATCH --error=/scratch/Users/maallen3/GATK/e_and_o/map.%j.err


module load hisat2
module load samtools

outdir=/scratch/Users/maallen3/GATK/map/
rootname=Eli
#rootname=Elizabeth
#rootname=Eric
#rootname=Ethan
infile1=chr21${rootname}Genome.end1.fastq
infile2=chr21${rootname}Genome.end2.fastq
indir=/scratch/Workshop/SR2019/8_GATK/reseq/

echo $rootname
echo $infile1
echo $indir

#pwd; hostname; date
#date

wc -l ${indir}/${infile1} >${outdir}lines_${rootname}.wc
mkdir -p ${outdir}sams/

date
#make a sam file

hisat2 --threads 4 --new-summary --very-sensitive \
        --no-spliced-alignment -x /scratch/Workshop/SR2019/8_GATK/hg38_GATK/HISAT/genome \
        --rg-id 3 \
        --rg PL:ILLUMINA \
        --rg PU:C6WMHACXX:3:none \
        --rg SM:${rootname} \
        -1 ${indir}/${infile1} -2 ${indir}/${infile2} \
        >${outdir}sams/${rootname}.sam
date

#create a bam file
mkdir -p ${outdir}bams/
samtools view -b ${outdir}sams/${rootname}.sam >${outdir}bams/${rootname}.bam
echo bam
date
#create a sorted bam file
mkdir -p ${outdir}sortedbams/
samtools sort -m 5G ${outdir}bams/${rootname}.bam >${outdir}sortedbams/${rootname}.sorted.bam
echo sorted.bam
date
samtools index ${outdir}sortedbams/${rootname}.sorted.bam
samtools flagstat ${outdir}sortedbams/${rootname}.sorted.bam >${outdir}sortedbams/${rootname}.sorted.bam.flagstat
echo indexed.bam
date

#--rg-id <lane>
#--rg SM:<person>
#--rg PL:<librarytype>
#--rg PU:{FLOWCELL_BARCODE}.{LANE}.{SAMPLE_BARCODE}.
#if you had several lanes of seq you would map each with different RG values and then merge the bam files using picard's merge
.
```

Step 2
Mark duplicates (requires data sorted by picardtools)

```
#!/bin/bash
#SBATCH --job-name=markdup  # Job name
#SBATCH --mail-type=ALL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=Mary.A.Allen@colorado.edu# Where to send mail
#SBATCH --nodes=1
#SBATCH --ntasks=4# Number of CPU (processer cores i.e. tasks) In this example I use 32 for bowtie2. I only need one, since no
ne of the commands I run are parallelized.
#SBATCH --time=00:10:00 # Time limit hrs:min:sec
#SBATCH -p compute
#SBATCH --mem=5gb # Memory limit
#SBATCH --output=/scratch/Users/maallen3/GATK/e_and_o/markdup.%j.out
#SBATCH --error=/scratch/Users/maallen3/GATK/e_and_o/markdup.%j.err


module load samtools

JAVAtemp=/scratch/Users/maallen3/tmp
mkdir -p $JAVAtemp
outdir=/scratch/Users/maallen3/GATK/map/
#rootname=Eli
#rootname=Elizabeth
#rootname=Eric
rootname=Ethan


INFILE=${outdir}sortedbams/${rootname}.sorted.bam
OUTDIR=${outdir}/remap/sortcord/
mkdir -p $OUTDIR
OUTFILE=${rootname}.bam

echo $INFILE
echo ${OUTDIR}${OUTFILE}
echo ${OUTDIR}${MarkdupOUTFILE}


#sort in the order GATK wants
java -Xmx5G -Djava.io.tmpdir=/scratch/Users/maallen3/tmp/${SLURM_JOBID} \
        -XX:ParallelGCThreads=4 -jar /opt/picard/2.6.0/picard-2.6.0.jar \
        SortSam INPUT=${INFILE} OUTPUT=${OUTDIR}$OUTFILE SORT_ORDER=coordinate

INFILE=${OUTDIR}$OUTFILE
OUTDIR=${outdir}/remap/markdup/
mkdir -p $OUTDIR
MarkdupOUTFILE=${rootname}markdup

#Mark dupicates in the bam as duplicate
java -Xmx5G -Djava.io.tmpdir=/scratch/Users/maallen3/tmp/${SLURM_JOBID} \
        -XX:ParallelGCThreads=4 -jar /opt/picard/2.6.0/picard-2.6.0.jar \
        MarkDuplicates INPUT=$INFILE OUTPUT=${OUTDIR}${OUTFILE} M=${OUTDIR}${MarkdupOUTFILE}


samtools index ${OUTDIR}${OUTFILE}



                                                                                                    54,0-1        Bot
```

Step 3
Realign and base recalibrate

Step 4
Create a gvcf for each person



Step 5
Merge the family
Step 6
Call snps (at a specific tranch)
Step 7
Call indels (at a specific tranch)
Step 8
Label potential de novo snps and indels (warning--- this is generally over calling)