

GATK has a ton of steps to call potential 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 (processor 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 (processor 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 duplicates 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}

```

Step 3

Realign and base recalibrate

```

allenma — maallen3@ip-172-31-7-94:~/GATK — ssh maallen3@3.17.49.74 — 171x48
|/bin/bash
#SBATCH --job-name=realignbaserecal # 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 (processor cores i.e. tasks) In this example I use 32 for bowtie2. I only need one, since none of the commands I run are parallelized.
#SBATCH --time=02:00:00 # Time limit hrs:min:sec
#SBATCH --p compute
#SBATCH --mem=5gb # Memory limit
#SBATCH --output=/scratch/Users/maallen3/GATK/e_and_o/realignbaserecal.%j.out
#SBATCH --error=/scratch/Users/maallen3/GATK/e_and_o/realignbaserecal.%j.err

outdir=/scratch/Users/maallen3/GATK/map/
JAVATemp=/scratch/Users/maallen3/tmp
mkdir -p $JAVATemp

hg38dir=/scratch/Workshop/SR2019/8_GATK/hg38_GATK/
#1000G_phase1.snps.high_confidence.hg38.vcf.gz
#dbsnp_146.hg38.vcf.gz
#Mills_and_1000G_gold_standard.indels.hg38.vcf.gz

INDIR=${outdir}/remap/markdup/
OUTDIR=${outdir}/remap/realign/
mkdir -p $OUTDIR

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

#find regions that need to be realigned (all your bam files need to be here!)
java -Xmx5G -Djava.io.tmpdir=/scratch/Users/maallen3/tmp/${SLURM_JOBID} -XX:ParallelGCThreads=4 -jar /opt/gatk/3.3-0/GenomeAnalysisTK.jar -T RealignerTargetCreator -nt 4 -R $(hg38dir)Homo_sapiens_assembly38.fasta -known $(hg38dir)Mills_and_1000G_gold_standard.indels.hg38.vcf.gz -known $(hg38dir)1000G_phase1.snps.high_confidence.hg38.vcf.gz -I $(INDIR)chr21EliGenome.bam -I $(INDIR)chr21EthanGenome.bam -I $(INDIR)chr21ElizabethGenome.bam -I $(INDIR)chr21EricGenome.bam -o ${OUTDIR}target_intervals.list

for infile in Eli.bam Eric.bam Elizabeth.bam Ethan.bam
do
realign those regions

java -Xmx5G -Djava.io.tmpdir=/scratch/Users/maallen3/tmp/${SLURM_JOBID} -XX:ParallelGCThreads=1 -jar /opt/gatk/3.3-0/GenomeAnalysisTK.jar -T IndelRealigner -R $(hg38dir)Homo_sapiens_assembly38.fasta -known $(hg38dir)Mills_and_1000G_gold_standard.indels.hg38.vcf.gz -known $(hg38dir)1000G_phase1.snps.high_confidence.hg38.vcf.gz -I $(INDIR)${infile} --targetIntervals ${OUTDIR}target_intervals.list -o ${OUTDIR}$(infile)
done

for infile in Eli.bam Eric.bam Elizabeth.bam Ethan.bam
do
INFILE=${OUTDIR}$(infile)
newOUTDIR=${outdir}/remap/baserecal/
mkdir -p $OUTDIR
BQSROUTFILE=${infile}.recaltable

#find regions that need to have base recalibration
java -Xmx5G -Djava.io.tmpdir=/scratch/Users/maallen3/tmp/${SLURM_JOBID} -XX:ParallelGCThreads=4 -jar /opt/gatk/3.3-0/GenomeAnalysisTK.jar -T BaseRecalibrator -nct 4 -o ${newOUTDIR}$(BQSROUTFILE) -I $(INFILE) -R $(hg38dir)Homo_sapiens_assembly38.fasta -knownSites $(hg38dir)Mills_and_1000G_gold_standard.indels.hg38.vcf.gz -knownSites $(hg38dir)1000G_phase1.snps.high_confidence.hg38.vcf.gz -knownSites $(hg38dir)dbsnp_146.hg38.vcf.gz

#recalibrate those regions
java -Xmx5G -Djava.io.tmpdir=/scratch/Users/maallen3/tmp/${SLURM_JOBID} -XX:ParallelGCThreads=4 -jar /opt/gatk/3.3-0/GenomeAnalysisTK.jar -T PrintReads --BQSR ${newOUTDIR}$(BQSROUTFILE) -nct 4 -o ${newOUTDIR}$(infile) -I $(INFILE) -R $(hg38dir)Homo_sapiens_assembly38.fasta

done

```

Step 4

Create a gvcf for each person

```
allenma — maallen3@ip-172-31-7-94:~/GATK — ssh maallen3@3.17.49.74 — 171x48
[maallen3@ip-172-31-7-94 GATK]$ vi 4_gvcf.sh
#!/bin/bash
#SBATCH --job-name=HaplotypeCaller # 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 (processor cores i.e. tasks) In this example I use 32 for bowtie2. I only need one, since none of the commands I run are parallelized.
#SBATCH --time=01:00:00 # Time limit hrs:min:sec
#SBATCH -p compute
#SBATCH --mem=5gb # Memory limit
#SBATCH --output=/scratch/Users/maallen3/GATK/e_and_o/HaplotypeCaller.%j.out
#SBATCH --error=/scratch/Users/maallen3/GATK/e_and_o/HaplotypeCaller.%j.err

outdir=/scratch/Users/maallen3/GATK/map/
JAVATemp=/scratch/Users/maallen3/tmp
mkdir -p $JAVATemp

hg38dir=/scratch/Workshop/SR2019/8_GATK/hg38_GATK/
INDIR=${outdir}/remap/baserecal/

for infile in Eli.bam Eric.bam Elizabeth.bam Ethan.bam
do
    INFILE=${INDIR}${infile}
    OUTDIR=${outdir}/remap/HaplotypeCaller/
    mkdir -p $OUTDIR

    java -Xmx5G -Djava.io.tmpdir=$JAVATemp/${SLURM_JOBID}/ -XX:ParallelGCThreads=4 -jar /opt/gatk/3.3-0/GenomeAnalysisTK.jar -T HaplotypeCaller -I $INFILE -o ${OUTDIR}${infile}
done
}.gvcf -R ${hg38dir}Homo_sapiens_assembly38.fasta -nct 4 --emitRefConfidence GVCF --dbsnp ${hg38dir}dbsnp_146.hg38.vcf.gz --variant_index_type LINEAR --variant_index_param
eter 128000

done
#Final time stamp
echo Time is `date`
```

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)