

Supplemental Code:

```
#!/bin/sh
#SBATCH --job-name=variant_calling_pipeline_mev-1 # Job name
#SBATCH --mail-type=ALL # Mail events (NONE, BEGIN, END, FAIL, ALL)
#SBATCH --mail-user=moeinraja@ufl.edu # Where to send mail
#SBATCH --cpus-per-task=8 # Number of CPU cores per task
#SBATCH --ntasks=1 # Run a single task
#SBATCH --mem=30gb # Memory limit
#SBATCH --time=85:55:00 # Time limit hrs:min:sec
#SBATCH --output=variant_calling_pipeline_mev-1_%j.out # Standard output and error log
#SBATCH --account=baer --qos=baer
```

step 1: Decompress the fq files:

```
gunzip *.fq.gz
```

replacing "_1.fq" with "R1.fq" in the fq files:

```
for i in *_1.fq; do
    mv "$i" "${i/%_1.fq/.R1.fq}";
done
```

```
for i in *_2.fq; do
    mv "$i" "${i/%_2.fq/.R2.fq}";
done
```

Step 2: Quality trimming

```
module load fastp
```

```
for i in *.R1.fq; do
    file=`basename $i .R1.fq`
    fastp --detect_adapter_for_pe \
    -i "$i" \
    -I "$file.R2.fq" \
    -o "${i/%.R1.fq/.R1_trimmed.fq}" \
    -O "${i/%.R1.fq/.R2_trimmed.fq}" \
    --html "${i/%.fq/.trimmed.html}" \
    --json "${i/%.fq/.trimmed.json}"
done
```

step 3: Bowtie2 alignment:

```
cp /ufrc/baer/moeinraja/reference_WS263/c_elegans.PRJNA13758.WS263.genomic.fa .
cp /ufrc/baer/moeinraja/reference_WS263/c_elegans.PRJNA13758.WS263.genomic.fa.fai .
cp /ufrc/baer/moeinraja/reference_WS263/c_elegans.PRJNA13758.WS263.genomic.dict .
```

```
module load gcc
```

```

module load bowtie2/
bowtie2-build c_elegans.PRJNA13758.WS263.genomic.fa ws263

for i in *.R1_trimmed.fq; do
    file=`basename $i .R1_trimmed.fq`
    bowtie2 -x ws263 -1 "$i" -2 "$file.R2_trimmed.fq" --phred33 -p 8 --very-sensitive-local -S
"$i/%.R1_trimmed.fq/.sam";
done

### step 4: Sam to Bam -- All alignments MQ<3/!PP are filtered out:

module load samtools/
module load bamtools/

for i in *.sam
do
    samtools view -b -q 3 -o "$i/%.sam/.bam" "$i"
done

rm *.sam

for i in *.bam
do
    samtools sort -O bam -T tmp_ -o "$i/%.bam/.output.bam" "$i"
done

for i in *.output.bam
do
    bamtools filter -isProperPair true -in "$i" -out "$i/%.output.bam/.PP.bam"
done

rm *.output.bam

### step 5: Add read group information using picard:

module load picard/2.18.20

for i in *.PP.bam; do
    x=`basename $i .PP.bam`
    picard AddOrReplaceReadGroups \
        I="$i" \
        O="$i/%.PP.bam/.PP.RG.bam" \
        RGID=HMH33DSXX.2 \
        RGLB="$x" \
        RGPL=illumina \
        RGPU=HMH33DSXX.2 \
        RGSM="$x"
done

```

```
rm *.PP.bam
```

step 6: Mark Duplicates:

```
for i in *.PP.RG.bam; do
java -Xmx8g -jar /apps/gatk/4.1.4.0/gatk-package-4.1.4.0-local.jar MarkDuplicates \
  -I "$i" \
  -O "${i/%.bam/.markduplicates.bam}" \
  -M "${i/%.bam/.markduplicates_metrics.txt}" \
  --CREATE_INDEX true \
  --REMOVE_DUPLICATES=true \
  --TMP_DIR /ufrc/baer/moeinraja/RIL/snp_test/shlee
done
```

```
rm *.PP.RG.bam
```

step 7: Call variants per-sample using HaplotypeCaller (in BP_RESOLUTION mode):

```
for i in *.PP.RG.markduplicates.bam; do
java -Xmx8g -jar /apps/gatk/4.1.4.0/gatk-package-4.1.4.0-local.jar HaplotypeCaller \
  -R c_elegans.PRJNA13758.WS263.genomic.fa \
  -I "$i" \
  -O "${i/%.PP.RG.markduplicates.bam/.bp.g.vcf.gz}" \
  -ERC BP_RESOLUTION
done
```

step 8: consolidating the vcf.gz files using GenomicsDBImport:

```
java -Xmx8g -jar /apps/gatk/4.1.4.0/gatk-package-4.1.4.0-local.jar GenomicsDBImport \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB1730.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB733.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB753.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB704.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB759.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB706.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB737.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB761.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB711.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB738.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB767.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB712.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB744.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB769.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB715.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB746.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB770.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB721.bp.g.vcf.gz \
```

```

-V pipelines_Charlie_mev-1_MA_FASTQ_CB748.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CBmev1.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB727.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB749.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB729.bp.g.vcf.gz \
-V pipelines_Charlie_mev-1_MA_FASTQ_CB752.bp.g.vcf.gz \
--genomicsdb-workspace-path mev-1_database_bp \
--intervals I --intervals II --intervals III --intervals IV --intervals V --intervals X --intervals MtDNA

```

step 9: Call variants jointly using GenotypeGVCFs in GATK:

```

java -Xmx8g -jar /apps/gatk/4.1.4.0/gatk-package-4.1.4.0-local.jar GenotypeGVCFs -R
c_elegans.PRJNA13758.WS263.genomic.fa -V gendb://mev-1_database_bp -O mev-
1_24samples.bp.GVCFs.vcf

```

step 10: Extract SNPs & INDELS:

```

for i in *.vcf; do
java -Xmx8g -jar /apps/gatk/4.1.4.0/gatk-package-4.1.4.0-local.jar SelectVariants \
-R /ufrc/baer/moeinraja/reference_WS263/c_elegans.PRJNA13758.WS263.genomic.fa \
-V "$i" \
--select-type-to-include SNP \
-O "${i}/%.vcf/.snp.vcf"
done

```

```

for i in *.vcf; do
java -Xmx8g -jar /apps/gatk/4.1.4.0/gatk-package-4.1.4.0-local.jar SelectVariants \
-R /ufrc/baer/moeinraja/reference_WS263/c_elegans.PRJNA13758.WS263.genomic.fa \
-V "$i" \
--select-type-to-include INDEL \
-O "${i}/%.vcf/.indel.vcf"
done

```

```

rm *.snp.indel.vcf
rm *.snp.indel.vcf.idx

```

step 11: Apply 3x-coverage filtering:

```

for i in *.vcf; do
java -Xmx8g -jar /apps/gatk/4.1.4.0/gatk-package-4.1.4.0-local.jar VariantFiltration \
-R /ufrc/baer/moeinraja/reference_WS263/c_elegans.PRJNA13758.WS263.genomic.fa \
-O "${i}/%.vcf/.3x.vcf" \
--variant "$i" \
--genotype-filter-name "DP" \
--genotype-filter-expression "DP < 3.0"
done

```

```

for i in *.3x.vcf; do
java -Xmx8g -jar /apps/gatk/4.1.4.0/gatk-package-4.1.4.0-local.jar SelectVariants \
-V "$i" \
--set-filtered-gt-to-nocall \
-O "${i}/%.3x.vcf/.3x.nocall.vcf}"
done

for i in *.3x.nocall.vcf; do
java -Xmx8g -jar /apps/gatk/4.1.4.0/gatk-package-4.1.4.0-local.jar SelectVariants \
-V "$i" \
-R /ufrc/baer/moeinraja/reference_WS263/c_elegans.PRJNA13758.WS263.genomic.fa \
-O "${i}/%.3x.nocall.vcf/.3x.filtered.vcf}" \
--exclude-filtered true \
--exclude-non-variants true
done

rm *.3x.vcf
rm *.3x.vcf.idx
rm *.nocall.vcf
rm *.nocall.vcf.idx

module load bcftools

for i in *.filtered.vcf; do
bcftools view --max-alleles 2 "$i" -o "${i}/%.vcf/.biallelic.vcf}"
done

# End

```