```
/*NEXTFLOW GATK PIPELINE FOR NYU DATE PALM GROUP*/
/* Ongoing adjusted pipeline (from production delete run pipeline, which was run in production run 11-6-21)
 * Added (so far [as of 1-22-22]): resource adjustments, checksum commands
 * [Production pipeline run, with delete of ALL big files (bams, [sample]_read_depth_out.txt, hap gcvf gzip file on work dir]
 * 1-22-22
 * M. Dhar < mid224@nyu.edu >
 *  Usage: nextflow run /path/to/main_index.nf
 */

// Setting some defaults here,
// can be overridden in config or via command line
params.out = "${params.outdir}/out"
params.tmpdir = "${params.outdir}/gatk_temp"

// Define modules here
BWA = 'bwa/intel/0.7.17'
PICARD = 'picard/2.17.11'
GATK = 'gatk/4.1.9.0'
R = 'r/intel/4.0.3'
SAMTOOLS = 'samtools/intel/1.11'

// Print paths
println "reads: $params.reads"
println "ref: $params.ref"
println "output: $params.out"
println "gatk temp dir: $params.tmpdir"

// Setup the reference file
ref = file(params.ref)

/* Read in files
*/
num_samples = 0
Channel
   .fromFilePairs( params.reads )
   .ifEmpty { error "Cannot find any reads matching: ${params.reads}"  }
   .tap { read_pairs_ch }
   .subscribe({ num_samples += 1 })

process align {

   input:
```

```
    set pair_id, file(reads) from read_pairs_ch

    output:
    set val(pair_id), file("${pair_id}_aligned_reads.bam"), env(task_dir) \
        into aligned_reads_ch

    script:
    readGroup = \
        "@RG\\tID:${pair_id}\\tLB:${pair_id}\\tPL:${params.pl}\\tSM:${pair_id}"
    """
    module load $BWA
    module load $SAMTOOLS
    bwa mem \
        -K 100000000 \
        -v 3 \
        -t ${task.cpus} \
        -Y \
        -R \"${readGroup}\" \
        $ref \
        ${reads[0]} \
        ${reads[1]} \
        | samtools view -b -@${task.cpus} -o ${pair_id}_aligned_reads.bam -

    task_dir="\$PWD"

    #get checksum of ${pair_id}_aligned_reads.bam
    chk=`md5sum ${pair_id}_aligned_reads.bam`
    echo "\$chk" >> "${params.outdir}/checksum.txt"

    """
}

process markDuplicatesSpark {
    publishDir "${params.out}/dedup_sorted", mode:'copy'

    input:
    set val(pair_id), file(aligned_reads), val(prev_task_dir) from aligned_reads_ch

    output:
    set val(pair_id), \
        file("${pair_id}_sorted_dedup.bam"), \
        env(task_dir) \
        into bam_for_variant_calling, \
        sorted_dedup_ch_for_metrics
```

```
    script:
    """
    module load $GATK
    gatk MarkDuplicatesSpark \
            -I $aligned_reads \
            -M ${pair_id}_dedup_metrics.txt \
            -O ${pair_id}_sorted_dedup.bam \
            --tmp-dir "/vast/mid224/tmp"

    task_dir="\$PWD"

    #get checksum of ${pair_id}_sorted_dedup.bam
    chk=`md5sum ${pair_id}_sorted_dedup.bam`
    echo "\$chk" >> "${params.outdir}/checksum.txt"

    #remove previous step's bams from work dir
    rm ${prev_task_dir}/${pair_id}_aligned_reads.bam

    """
}

process getMetrics{
    publishDir "${params.out}/metrics", mode:'copy'

    input:
    set val(pair_id), \
            file(sorted_dedup_reads), \
            val(prev_task_dir) \
            from sorted_dedup_ch_for_metrics

    output:
            set val(pair_id), \
                file("${pair_id}_alignment_metrics.txt"), \
                file("${pair_id}_insert_metrics.txt"), \
                file("${pair_id}_insert_size_histogram.pdf"), \
            file("${pair_id}_depth_out.txt")
            set val(pair_id), \
                        env(task_dir) \
                        into info_for_clean_metrics_ch

    script:
    """
    module load $PICARD
```

```
    module load $R
    module load $SAMTOOLS
    java -jar \$PICARD_JAR \
       CollectAlignmentSummaryMetrics \
          R=${params.ref} \
       I=${sorted_dedup_reads} \
          O=${pair_id}_alignment_metrics.txt
    java -jar \$PICARD_JAR \
       CollectInsertSizeMetrics \
       INPUT=${sorted_dedup_reads} \
          OUTPUT=${pair_id}_insert_metrics.txt \
       HISTOGRAM_FILE=${pair_id}_insert_size_histogram.pdf
    samtools depth -a ${sorted_dedup_reads} > ${pair_id}_depth_out.txt

    task_dir="\$PWD"

    #get checksum of ${pair_id}_alignment_metrics.txt
    chk1=`md5sum ${pair_id}_alignment_metrics.txt`
    echo "\$chk1" >> "${params.outdir}/checksum.txt"

    #get checksum of ${pair_id}_insert_metrics.txt
    chk2=`md5sum ${pair_id}_insert_metrics.txt`
    echo "\$chk2" >> "${params.outdir}/checksum.txt"

    #get checksum of ${pair_id}_insert_size_histogram.pdf
    chk3=`md5sum ${pair_id}_insert_size_histogram.pdf`
    echo "\$chk3" >> "${params.outdir}/checksum.txt"

    #get checksum of ${pair_id}_depth_out.txt
    chk4=`md5sum ${pair_id}_depth_out.txt`
    echo "\$chk4" >> "${params.outdir}/checksum.txt"

    """
}

process cleanMetrics {

    input:
      set val(pair_id), \
        val(prev_task_dir) \
        from info_for_clean_metrics_ch

    script:
    """
```

```
    rm ${prev_task_dir}/${pair_id}_depth_out.txt
    """
}

process haplotypeCaller {
    publishDir "${params.out}/haplotype_caller", mode:'copy'

    input:
    set val(pair_id), \
      file(input_bam), \
      val(prev_task_dir) \
      from bam_for_variant_calling

    output:
    set val(pair_id), file("${pair_id}_raw_variants.g.vcf.gz"), \
            env(task_dir) \
            into hc_output_ch

    script:
    """
    module load $GATK
    gatk HaplotypeCaller \
      -R $ref \
      -I $input_bam \
      -O ${pair_id}_raw_variants.g.vcf.gz \
      -ERC GVCF

    #get checksum of ${pair_id}_raw_variants.g.vcf.gz
    chk=`md5sum ${pair_id}_raw_variants.g.vcf.gz`
    echo "\$chk" >> "${params.outdir}/checksum.txt"

    rm ${prev_task_dir}/${pair_id}_sorted_dedup.bam
    task_dir="\$PWD"

    """

}

process IndexFeatureFile {
    publishDir "${params.out}/haplotype_caller", mode:'copy'

    input:
            set val(pair_id), \
              file(input_gvcf), \
```

```
            val(prev_task_dir) \
            from hc_output_ch

    output:
        file("${input_gvcf}.tbi")

    script:
    """
    module load $GATK
    gatk IndexFeatureFile \
        -I $input_gvcf

    rm ${prev_task_dir}/${pair_id}_raw_variants.g.vcf.gz

    #get checksum of ${input_gvcf}.tbi
    chk=`md5sum ${input_gvcf}.tbi`
    echo "\$chk" >> "${params.outdir}/checksum.txt"

    """
}
```

#NEXTFLOW GATK PIPELINE SBATCH SCRIPT

```
#!/bin/bash
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --cpus-per-task=1
#SBATCH --time=168:00:00
#SBATCH --mem=2GB
#SBATCH --job-name=ongoing_adjusted_prod_run_delete
#SBATCH --mail-type=END
#SBATCH --mail-user=mid224@nyu.edu
#SBATCH --output=slurm_%j.out

module purge
module load nextflow/20.11.0-edge

#Run local script with resume and report
nextflow main_ongoing_adjusted.nf -with-report report_slurm_%j.html -resume

sort -k 2 out/checksum.txt > out/checksum_sorted.txt
```

```
//NEXTFLOW GATK PIPELINE CONFIG FILE

// Required Parameters
//FULL PARAMS READS
//params.reads = "$SCRATCH/data/RSYNC/FastQ_Samples_9-19-
21/Sym_Links/*/*{1,2}.fastq.gz"
//2-SAMPLE PARAMS READS
params.reads = "$SCRATCH/data/sample_data/*/*{1,2}.fastq.gz"
//params.ref =
"/scratch/mid224/data/genome_assemblies/tiago_new/concat/bc4male_v2.fasta"
params.ref = "$SCRATCH/data/genome_assemblies/tiago_new/concat/bc4male_v2.fasta"
params.outdir = "$SCRATCH/prod_adjust/ongoing_adjusted_prod_del/out"
params.pl = "ILLUMINA"

// Set the Nextflow Working Directory
// By default this gets set to params.outdir + '/nextflow_work_dir'
workDir = params.outdir + '/nextflow_work_dir'


//NEW RESOURCE ALLOCATIONS 1-22-22
// slurm and resource allocation
process {
   executor = 'slurm'
   cpus = 10
   memory = { 15.GB * task.attempt }
   time = { 30.min * task.attempt }
   withName: getMetrics { cpus = '1' }
   withName: haplotypeCaller { cpus = '1' }
   withName: IndexFeatureFile { cpus = '1' }
   withName: getMetrics { memory = '1.GB' }
   withName: cleanMetrics { memory = '16.MB' }
   withName: haplotypeCaller { memory = '2.GB' }
   withName: IndexFeatureFile { memory = '4.GB' }
   withName: align { time = '10.h' }
   withName: haplotypeCaller { time = '100.h' }
   withName: markDuplicatesSpark { time = '10.h' }
   withName: getMetrics { time = '3.h' }
   withName: cleanMetrics { time = '10.min' }
   withName: IndexFeatureFile { time = '15.min' }
}
```