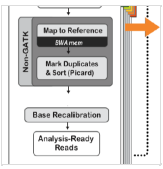


Dec 15, 2020

High-throughput SARS-CoV-2 and host genome sequencing - alignment protocol

In 1 collection

Christopher Hughes[1]

[1]Stanford University

In Development This protocol is published without a DOI.

Christopher Hughes

ABSTRACT

This protocol is designed to perform alignment on low-coverage (0.5x - 3x) whole-genome human sequences from Illumina MiSeq and NovaSeq platforms. This protocol was derived from two previously validated alignment and variant calling pipelines:

Nextflow:
nf-core/sarek 2.5.2

https://github.com/nf-core/sarek
Garcia M, Juhos S, Larsson M et al. Sarek: A portable workflow for whole-genome sequencing analysis of germline and somatic variants [version 1; peer review: awaiting peer review]. *F1000Research* 2020, 9:63. doi: 10.12688/f1000research.16665.1.

and

WDL + GATK4:
gatk4-genome-processing-pipeline
https://github.com/gatk-workflows/gatk4-genome-processing-pipeline
Poplin, R. *et al.* Scaling accurate genetic variant discovery to tens of thousands of samples. *bioRxiv* 201178 (2017). doi:10.1101/201178

Protocol image adapted (removed indel realignment step) from Figure 1 of Poplin, R. et al (2017).

What this pipeline does:
1. Performs quality check of demultiplexed FASTQ data
2. Aligns and maps FASTQ to the human reference
3. Marks duplicates in the alignments
4. Recalibrates base quality using known variants
5. Creates an index for the .BAM file
6. Validates the .BAM file for any errors
7. Calculates coverage per contig, .BAM file statistics, and plots those statistics

Inputs:
Demultiplexed FASTQ files for a single sample for one or multiple lanes

Outputs:
Aligned, de-deduped, recalibrated, QC'd BAM file
Coverage statistics and plots
BAM statistics and plots
FASTQ statistics and plots

PROTOCOL CITATION

Christopher Hughes 2020. High-throughput SARS-CoV-2 and host genome sequencing - alignment protocol . **protocols.io**
https://protocols.io/view/high-throughput-sars-cov-2-and-host-genome-sequenc-bf4mjqu6

COLLECTIONS

COVID19

KEYWORDS

Whole-genome sequencing alignment

CREATED

May 06, 2020

LAST MODIFIED

Dec 15, 2020

PROTOCOL INTEGER ID

36717

PARENT PROTOCOLS

Part of collection
COVID19

GUIDELINES

As always, be cognizant of protecting patient information where applicable. Please review any pertitent IRB protocols, patient consent forms, HIPAA guidelines, and bioethics principles.

SAFETY WARNINGS

This protocol attempts to follow and implement the "Best Practices" guidelines of GATK4. It should be noted that these practices may change with the advent of newer technologies and methods. In the future, "Best Practices" may not be commensurate with this pipeline and it is the user's responsibility to validate the most current techniques prior to use.

This pipeline does not perform indel realignment as this preprocessing step was retired in GATK 3.6. Rationale for this decision can be read here:

https://github.com/broadinstitute/gatk-docs/blob/master/blog-2012-to-2019/2016-06-21-Changing_workflows_around_calling_SNPs_and_indels.md?id=7847

ABSTRACT

This protocol is designed to perform alignment on low-coverage (0.5x - 3x) whole-genome human sequences from Illumina MiSeq and NovaSeq platforms. This protocol was derived from two previously validated alignment and variant calling pipelines:

Nextflow:
nf-core/sarek 2.5.2

https://github.com/nf-core/sarek
Garcia M, Juhos S, Larsson M et al. Sarek: A portable workflow for whole-genome sequencing analysis of germline and somatic variants [version 1; peer review: awaiting peer review]. *F1000Research* 2020, 9:63. doi: 10.12688/f1000research.16665.1.

and

WDL + GATK4:
gatk4-genome-processing-pipeline
https://github.com/gatk-workflows/gatk4-genome-processing-pipeline

Poplin, R. *et al.* Scaling accurate genetic variant discovery to tens of thousands of samples. *bioRxiv* 201178 (2017). doi:10.1101/201178

Protocol image adapted (removed indel realignment step) from Figure 1 of Poplin, R. et al (2017).

What this pipeline does:
1. Performs quality check of demultiplexed FASTQ data
2. Aligns and maps FASTQ to the human reference
3. Marks duplicates in the alignments
4. Recalibrates base quality using known variants
5. Creates an index for the .BAM file
6. Validates the .BAM file for any errors
7. Calculates coverage per contig, .BAM file statistics, and plots those statistics

Inputs:
Demultiplexed FASTQ files for a single sample for one or multiple lanes

Outputs:
Aligned, de-deduped, recalibrated, QC'd BAM file
Coverage statistics and plots
BAM statistics and plots
FASTQ statistics and plots

BEFORE STARTING

Before running any bioinformatic pipelines, it's appropriate to verify the software versions used. For this protocol we used:

System:
Linux 3.10.0-957.27.2.el7.x86_64 x86_64
-NAME="CentOS Linux"
-VERSION="7 (Core)"
-ID="centos"
-ID_LIKE="rhel fedora"
-VERSION_ID="7"
-PRETTY_NAME="CentOS Linux 7 (Core)"
-ANSI_COLOR="0;31"
-CPE_NAME="cpe:/o:centos:centos:7"
-HOME_URL="https://www.centos.org/"
-BUG_REPORT_URL="https://bugs.centos.org/"
-CENTOS_MANTISBT_PROJECT="CentOS-7"
-CENTOS_MANTISBT_PROJECT_VERSION="7"
-REDHAT_SUPPORT_PRODUCT="centos"
-REDHAT_SUPPORT_PRODUCT_VERSION="7"

SLURM 19.05.6
Modules Version 8.3.4

Alignment:
The Genome Analysis Toolkit (GATK) v4.1.4.1
-HTSJDK Version: 2.21.0
-Picard Version: 2.21.2
-MarkDuplicates
-BaseRecalibrator
-ApplyBQSR

Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
-Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)

Burrows-Wheeler transformation Version: 0.7.17-r1188

FastQC v0.11.8

Samtools Version: 1.8 (using htslib 1.8)
-plot-bamstats
-GNUPLOT Version 5.2 patchlevel 0
Samtools Version: 1.10 (using htslib 1.10)

Mosdepth 0.2.3

QualiMap v.2.2.1

The genomic references and variants used for alignment, recalibration, and variant calling are listed below:

Human reference:
UCSC Genome Browser assembly ID: hg38
Sequencing/Assembly provider ID: Genome Reference Consortium Human GRCh38.p12 (GCA_000001405.27)
Assembly date: Dec. 2013 initial release; Dec. 2017 patch release 12
Assembly accession:GCA_000001405.27
NCBI Genome ID:51 (Homo sapiens (human))
NCBI Assembly ID:5800238 (GRCh38.p12, GCA_000001405.27)
BioProject ID:PRJNA31257

SARS-CoV-2 reference:
LOCUS NC_045512 29903 bp ss-RNA linear VRL 30-MAR-2020
DEFINITION Severe acute respiratory syndrome coronavirus 2 isolate Wuhan-Hu-1, complete genome.
ACCESSION NC_045512
VERSION NC_045512.2
DBLINK BioProject: PRJNA485481
KEYWORDS RefSeq.
SOURCE Severe acute respiratory syndrome coronavirus 2 (SARS-CoV2)

The reference used for alignment in this project contains both hg38 and the SARS-CoV-2 contig appended to the end of the fasta. The reference pathway is below alongside the appropriate indexes.

Recalibration known-sites:
We used three known-variant lists to perform base recalibration. These VCFs were obtained from the Broad's GATK Resource Bundle FTP:

ftp://ftp.broadinstitute.org/bundle/hg38/

This FTP can be accessed anonymously. Users will be prompted for a password to access the FTP, however, leave the password field blank to authenticate.

We used:
1000G_phase1.snps.high_confidence.hg38.vcf.gz
dbsnp_146.hg38.vcf.gz
Mills_and_1000G_gold_standard.indels.hg38.vcf.gz

Sequence Data:
This protocol uses two sets of sequencing data from 160 COVID-19 patients. One set of sequencing data was derived from the Illumina's MiSeq platform as a quality-control check and other set was derived from NovaSeq which was used in the analyses.

Alignment

1 The pipeline is broken into two steps: alignment and recalibration. First, alignment is performed using the alignment.sh script:

alignment.sh

```
#!/bin/bash

#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=8
#SBATCH --account=euan
#SBATCH --partition=batch
#SBATCH --time=20:00:00
#SBATCH --mail-user=email@email.edu --mail-type=BEGIN,END,FAIL
#SBATCH --mem=16G

#Load tools
ml purge
ml bwa
ml samtools
ml fastqc


#Load sample name
R1=$1
R2=$2
sample_name=$3
lane=$4
echo $R1 $R2 $sample_name $lane


#Step 0: Perform FASTQ QC and output reports for R1/R2
fastqc -t 2 ${R1} -o /path/to/${sample_name}/fastqc
fastqc -t 2 ${R2} -o /path/to/${sample_name}/fastqc

#Step 1: Assign read groups and align
read_tmp=$(zcat ${R1} | grep ^@ | head -1 | sed 's|\:|\t|g')

readGroup="@RG\\tID:$(echo $read_tmp | awk '{print $3 "."
$4}')\\tCN:Stanford\\tSM:${sample_name}\\tLB:${sample_name}\\tPL:illumina"

#Alignment
bwa mem -K 100000000 \
-R ${readGroup} \
-t ${threads} \
-M \
${human_reference} \
${R1} \
${R2} \
-o ${sample_name}_${lane}.bam

#Sort the aligned .bam
samtools sort --threads -@ ${threads} -m ${memory} \
${sample_name}/${sample_name}_${lane}.bam \
```

```
-o ${sample_name}_${lane}.bam

#Step 2: Index BAM
samtools index -@ ${threads} ${sample_name}_${lane}.bam
```

Performs BWA alignment of a sample and produces QC reports of demultiplex FASTQ files

CentOS Linux 7 (Core)

Recalibration

2 After each lane for the sample has been aligned, the .bams are merged, duplicates are marked and base recalibration is performed using recalibration.sh:

recalibration.sh

```
#!/bin/bash

#SBATCH --nodes=1
#SBATCH --ntasks=1
#SBATCH --cpus-per-task=8
#SBATCH --account=euan
#SBATCH --partition=batch
#SBATCH --time=16:00:00
#SBATCH --mail-user=email@email.edu --mail-type=BEGIN,END,FAIL
#SBATCH --mem=16G

#Load tools
ml purge
ml samtools
ml gatk
ml qualimap
ml mosdepth

#Load sample name
sample_name=$1

cd /path/to/${sample_name}/


#Merges each lane for the sample where L* is a wild card for L001 or L002
samtools merge -@ ${threads} -f ${sample_name}_merged.bam
${sample_name}_L*.bam
samtools index -@ ${threads} ${sample_name}_merged.bam

#Mark Duplicates
gatk --java-options "-Xmx12G -XX:ConcGCThreads=1" \
MarkDuplicates \
--USE_JDK_DEFLATER true \
--USE_JDK_INFLATER true \
--MAX_RECORDS_IN_RAM 50000 \
```

```
--INPUT ${sample_name}_merged.bam \
--METRICS_FILE ${sample_name}.bam.metrics \
--TMP_DIR . \
--CREATE_INDEX true \
--OUTPUT ${sample_name}.md.bam


#Removes the merged BAM file to keep things cleaned up if disk space is a
concern
if [ -s ${sample_name}.md.bam ]; then
 echo Cleaning up intermediate BAM files...
 rm ${sample_name}_merged.ba*
fi

#Create base recalibration table
gatk --java-options "-Xmx12G -XX:ConcGCThreads=1" \
BaseRecalibrator \
-I ${sample_name}.md.bam \
-O ${sample_name}.recal.table \
-R ${human_reference} \
--use-jdk-deflater true \
--use-jdk-inflater true \
--known-sites /path/to/Mills_and_1000G_gold_standard.indels.hg38.vcf.gz \
--known-sites /path/to/dbsnp_146.hg38.vcf.gz \
--known-sites /path/to/hg381000G_phase1.snps.high_confidence.hg38.vcf.gz

#Apply base recalibration table
gatk --java-options "-Xmx12G -XX:ConcGCThreads=1" \
ApplyBQSR \
-R ${human_reference} \
--input ${sample_name}.md.bam \
--output ${sample_name}.recal.bam \
--bqsr-recal-file ${sample_name}.recal.table

#Index the recalibrated BAM
samtools index -@ ${threads} ${sample_name}.recal.bam

#Allows for sanity check on the .bam prior to generating metrics
echo Applying Samtools Quickcheck...
samtools quickcheck ${sample_name}.recal.bam >
${sample_name}.quickcheck.txt

#Validate the SAM for other issues
gatk --java-options "-Xmx12G -XX:ConcGCThreads=1" \
ValidateSamFile \
-I ${sample_name}.recal.bam \
--MODE SUMMARY \
-O ${sample_name}.validateSAM.txt

#Generate metrics for BAM
samtools stats ${sample_name}.recal.bam > ${sample_name}.samtools.stats.out
plot-bamstats -p ${sample_name}.plots ${sample_name}.samtools.stats.out


#Clean up and rename
mv ${sample_name}.recal.bam ${sample_name}_${batch_name}.bam
```

```
mv ${sample_name}.recal.bam.bai ${sample_name}_${batch_name}.bai

mkdir /path/to/${sample_name}/intermediates
mv ${sample_name}.* intermediates/
mv ${sample_name}_L* intermediates/
mv ${sample_name}_merged* intermediates/

#Calculate Coverage
mkdir path/to/${sample_name}/${sample_name}
unset DISPLAY

qualimap bamqc -outdir path/to/${sample_name}/${sample_name}/ -outformat
HTML -bam ${sample_name}_${batch_name}.bam


mosdepth -t ${threads} path/to/${sample_name}/${sample_name}/
${sample_name}_${batch_name}.bam
```

Performs BAM merging, duplicate marking, recalibration, and statistics on the BAMs

CentOS Linux 7 (Core)

SLURM Submission

3 An example script of how each sample is submitted to SLURM:

sbatch command for alignment

```
sbatch --job-name=${sample_name}_${lane}_alignment \
--output=/path/to/${sample_name}/${sample_name}_${lane}_pipeline.log \
/path/to/alignment.sh \
/path/to/${sample_name}_${lane}_R1.fastq.gz \
/path/to/${sample_name}_${lane}_R2.fastq.gz \
${sample_name} \
${lane}
```

Example code of how samples are submitted to SLURM for alignment

CentosOS Linux 7 (Core)

sbatch command for recalibration

```
sbatch --job-name=${sample_name}_recalibration \
--output=/path/to/${sample_name}/${sample_name}_recalibration.log \
/path/to/recalibration.sh \
${sample_name}
```

Example code of how samples are submitted to SLURM for recalibration

CentosOS Linux 7 (Core)

MultiQC Report

4 MultiQC is a tool that allows for aggregation of several output reports from each sample. We use MultiQC to generate reports for all of our samples by navigating to the parent directory and running the command:

multiqc

```
cd /path/to/parent/of/samples/

multiqc . --interactive
```

Run multiqc on all sample directories

CentosOS Linux 7 (Core)

4.1 A configuration file is used to generate the MultiQC report. This .yaml should be saved in the home directory as multiqc_config.yaml:

multiqc_config.yaml

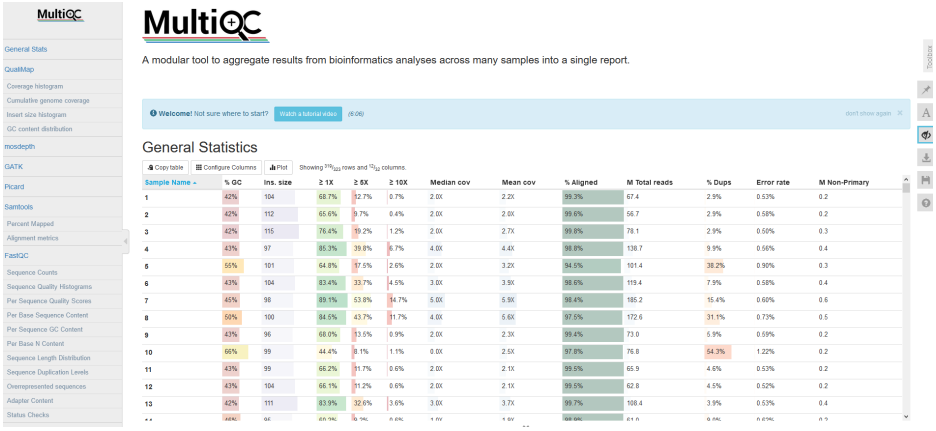
```
extra_fn_clean_exts:
   - .clean
   - _S
   - _final
   - .AHJHNYCCXX
   - _merged

table_columns_visible:
 QualiMap:
  avg_gc: True
  median_insert_size: True
  1_x_pc: True
  5_x_pc: True
  10_x_pc: True
  30_x_pc: False
  median_coverage: True
  mean_coverage: True
  percentage_aligned: True
  total_reads: True

 mosdepth:
  median_coverage: False
  1_x_pc: False
  5_x_pc: False
  10_x_pc: False
  30_x_pc: False
  50_x_pc: False

 Samtools:
  error_rate: True
  non-primary_alignments: True
  reads_mapped: False
  reads_mapped_percent: False
```

```
  reads_properly_paired_percent: False
  reads_MQ0_percent: False
  raw_total_sequences: False

 FastQC:
  percent_duplicates: False
  percent_gc: False
  avg_sequence_length: False
  percent_fails: False
  total_sequences: False

custom_plot_config:
 mosdepth-coverage-per-contig:
  ymax: 20

top_modules:
 - 'qualimap'

remove_sections:
 - mosdepth-coverage-dist
 - mosdepth-coverage-cov
```

The configuration file used to generate the MultiQC report.

CentosOS Linux 7 (Core)

Expected output:



Example report generated from MultiQC using the command and configuration above.