**Input format**

* **Two files per sample, ending with**

\_R1.fastq.gz (forward read)

\_R2.fastq.gz (reverse read)

Or

\_L[0-9]\*\_R1\_001.fastq.gz

\_L[0-9]\*\_R2\_001.fastq.gz

In this case the samples will be concatenated.

If there is a mix of R1.fastq.gz and \_L[0-9]\*\_R1\_001.fastq.gz, samples will not be concatenated and the pipeline will generate an error. *Can be improved to fix this*

The arguments for the pipeline:

* the path to the folder where **the input files** are located
* the path to the folder for the **output files**
* job\_name: this is the name that will be in all output files;

sbatch /path/to/the/input/files /path/to/the/output *job\_name*

It will output the following folder structure, if the job\_name is test:

test\_results

test\_fastqc

test\_bam

test\_vcf

**test\_report**

test\_logs

test\_rawfiles

The summary of the output are in **job\_name\_report** folder (e,g. test\_report):

**SUMMARY OF QUALITY CONTROLS that should be checked and taken into consideration before looking at vcf files:**

1. Fastqc:

– number of raw reads;

- average quality of the reads;

- average GC content

2) Number of reads aligned (bwa output?)

70%+

3) % of duplicates. The high number of duplicates is expected when the total number of reads is high; nothing to worry if the coverage is high; the multiplication rate of reads should be evenly distributed.

4) Coverage of exome: what percentage of exome is covered with more than 30 reads - how comprehensive the variant calling will be as the

(5) Coverage of MitoCarta exons.

6) Contamination as judged by verifybamid and haplotype contamination screening using haplocheck.

**PIPELINE OUTLINE (as of Oct 2021)**

**1. Alignment**  
bwa mem aligner; default parameters.

Alignment to the GRC38 genome without alt contigs (GCA\_000001405.15\_GRCh38\_no\_alt\_analysis\_set)

**2. Removing duplicates**

Picard MarkDuplicates

**3. Variant calling**

GATK pipeline (v4.1.9).

(Call variants for every nucleotide of a single file (\*g.vcf files); combine the files, call variants on all samples.

The hard filters are implemented separately for the indels and SNPs.

For SNPs:

-filter "QD < 2.0" --filter-name "QD2" \

-filter "QUAL < 30.0" --filter-name "QUAL30" \

-filter "SOR > 3.0" --filter-name "SOR3" \

-filter "FS > 60.0" --filter-name "FS60" \

-filter "MQ < 40.0" --filter-name "MQ40" \

-filter "MQRankSum < -12.5" --filter-name "MQRankSum-12.5" \

-filter "ReadPosRankSum < -8.0" --filter-name "ReadPosRankSum-8"

For indels:

-filter "QD < 2.0" --filter-name "QD2" \

-filter "QUAL < 30.0" --filter-name "QUAL30" \

-filter "FS > 200.0" --filter-name "FS200" \

-filter "ReadPosRankSum < -20.0" --filter-name "ReadPosRankSum-20"

Note the deviations from the GATK Best practices:

* No BQSR quality recalibration, as doesn’t produce a noticebale improvement but is computationally expensive (e.g., Kobolt (2020). Best practices for variant calling in clinical sequencing. BMC Genome Medicine).
* No VQSR for variant filtering as unclear how it works, how to choose the parameters and benchmarking has shown inferior performance of it, hard filtering instead.

Then normalisation of the variants is using bcftools norm.

**4. Variant annotation**

VEP104, ensemble annotation; plus filtering using python script.

Plugins in use: CADD, dbNSFP, ExACpLI, loFtool, DisGeNET, REVEL, Mastermind.

**Usability**:

To move the scripts to another location, update the script path location in master\_exome\_DD.sh

E.g., to

script\_path="/mnt/nfs/home/ndd73/exomes\_pipeline/exomes\_pipeline"

All the required plugins/software are installed in rtmngs account.

**Updates of the software/plugins/datasets potentially required:**

VEP (current version 104) and the datasets associated with plugins, if required;

GATK – update to DRAGEN\_GATK

OMIM dataset (the current one is based on April 2021)