# Before the code…

Hi! If you are a QCB student, good luck with the exam! If you need any help, feel free to reach out. You can use this code pipeline as a reference to check whether you’ve completed everything that was requested. Just make sure to first read the *CHG\_Assignment\_24\_25.pdf* on the GitHub page to see if the professors have updated any requirements.

**Note:** The final results of my project were not the best. Based on my experience, here are two suggestions:

1. Avoid adding supplementary material.
2. Take a look at the GitHub pages of my colleagues whose projects were more appreciated by the professors.

Thanks for visiting my GitHub page!

# Initial Analysis and Q60 filtering

**THE SAME LINES OF CODE MUST ALSO BE DONE FOR THE CONTROL.BAM FILE**

samtools sort Tumor.bam > Tumor.sorted.bam

samtools view -b -h -q 60 -F 260 Tumor.sorted.bam > **Tumor.sorted.q60.bam**

samtools index Tumor.sorted.q60.bam

samtools flagstat Tumor.sorted.q60.bam

samtools stats Tumor.sorted.q60.bam > Tumor.stats.q60.txt

samtools bedcov Captured\_Regions.bed Tumor.sorted.q60.bam > Tumor.bedcov.q60.txt

samtools bedcov DNA\_Repair\_Genes.bed Tumor.sorted.q60.bam > Tumor.bedcov.repair.q60.txt

# Realignment

java -jar GenomeAnalysisTK.jar -T RealignerTargetCreator -R human\_g1k\_v37.fasta -I Tumor.sorted.q60.bam -o Tumor.realigner.intervals -L Captured\_regions.bed

java -jar GenomeAnalysisTK.jar -T IndelRealigner  -R human\_g1k\_v37 -I Tumor.sorted.q60.bam -targetIntervals Tumor.realigner.intervals -o **Tumor.realigned.q60.bam** -L Captured\_regions.bed

samtools view Tumor.realigner.q60.bam | grep OC | wc -l

samtools view Control.realigner.q60.bam | grep OC | wc -l

# Recalibration

java -jar GenomeAnalysisTK.jar -T BaseRecalibrator -R human\_g1k\_v37.fasta -I Tumor.realigned.q60.bam -knownSites hapmap\_3.3.b37.vcf -o Tumor.recal.table -L Captured\_regions.bed

java -jar GenomeAnalysisTK.jar -T PrintReads -R human\_g1k\_v37.fasta -I Tumor.realigned.q60.bam -BQSR Tumor.recal.table -o **Tumor.recal.q60.bam** -L Captured\_regions.bed

java -jar GenomeAnalysisTK.jar -T BaseRecalibrator -R human\_g1k\_v37.fasta -I Tumor.realigned.q60.bam -knownSites hapmap\_3.3.b37.vcf -BQSR Tumor.recal.table -o Tumor.after.recal.table -L Captured\_regions.bed

java -jar GenomeAnalysisTK.jar -T AnalyzeCovariates -R human\_g1k\_v37.fasta -before Tumor.recal.table -after Tumor.after.recal.table -plots Tumor.recal.plots.pdf -csv Tumor.recal.csv

samtools view Tumor.recal.q60.bam | grep OC | wc -l

samtools view Tumor.recal.q60.bam | grep OQ | wc -l

samtools view Control.recal.q60.bam | grep OC | wc -l

samtools view Control.recal.q60.bam | grep OQ | wc -l

# Deduplication

java -jar picard.jar MarkDuplicates I=Tumor.recal.q60.bam O=**TumorGOD.bam** REMOVE\_DUPLICATES=true TMP\_DIR=/tmp METRICS\_FILE=Sample.picard.log ASSUME\_SORTED=true

samtools index **TumorGOD.bam**

**FROM HERE WE WILL OMIT THE Q60 FROM THE NAME**

samtools stats TumorGOD.bam > TumorGOD.stats.txt

samtools bedcov Captured\_Regions.bed TumorGOD.bam > TumorGOD.bedcov.txt

samtools bedcov DNA\_Repair\_Genes.bed TumorGOD.bam > TumorGOD.bedcov.repair.txt

# Variant calling (bcftools)

bcftools mpileup -Ou -a DP -f human\_g1k\_v37.fasta TumorGOD.bam | bcftools call -Ov -c -v > TumorGOD.bcf.vcf

vcftools –minQ 20 –max-meanDP 200 –min-meanDP 5 –remove-indels –vcf TumorGOD.bcf.vcf –out TumorGOD.bcf.filtered –recode –recode-INFO-all

**THIS WILL ALSO CREATE “TumorGOD.bcf.filtered.recode.vcf”**

# Variant Annotation

**SNP Eff**

java -Xmx4g -jar snpEff.jar -v hg19kg TumorGOD.bcf.filtered.recode.vcf -s TumorGOD.bcf.filtered.recode.vcf.html > TumorGOD.bcf.filtered.recode.ann.vcf

**SNP Sift**

java -Xmx4g -jar SnpSift.jar Annotate hapmap\_3.3.b37.vcf TumorGOD.bcf.filtered.recode.ann.vcf > TumorGOD.bcf.filtered.recode.ann.hapmap.vcf

java -Xmx4g -jar SnpSift.jar Annotate clinvar\_Pathogenic.vcf TumorGOD.bcf.filtered.recode.ann.hapmap.vcf > TumorGOD.bcf.filtered.recode.ann.pathogenic.vcf

**FILTERING**

java -Xmx4g -jar SnpSift.jar filter "((exists CLNSIG) || ((ANN[ANY].IMPACT = 'HIGH') || (ANN[ANY].IMPACT = 'MODERATE') || ((ANN[ANY].IMPACT = 'LOW') && (ANN[ANY].EFFECT has 'splice\_region\_variant')) || ((ANN[ANY].IMPACT = 'MODIFIER') && ((ANN[ANY].EFFECT has '5\_prime\_UTR\_variant') || (ANN[ANY].EFFECT has '3\_prime\_UTR\_variant') || (ANN[ANY].EFFECT has 'regulatory\_region\_variant') || (ANN[ANY].EFFECT has 'TF\_binding\_site\_variant'))))) && (DP > 20) && (exists ID)" < TumorGOD.bcf.filtered.recode.ann.pathogenic.vcf > TumorGOD.bcf.filtered.recode.ann.pathogenic.SUPER\_FILTER.vcf

# Somatic variant calling (mpileup + VarScan)

samtools mpileup -q 1 -f human\_g1k\_v37.fasta TumorGOD.bam > TumorGOD.pileup

samtools mpileup -q 1 -f human\_g1k\_v37.fasta ControlGOD.bam > ControlGOD.pileup

java -jar VarScan.v2.3.9.jar somatic ControlGOD.pileup TumorGOD.pileup –output-snp [somatic.pm](http://somatic.pm) –ouput-indel somatic.indel –output-vcf 1

# Somatic variant annotation

vcftools --min-meanDP 30 --remove-indels --vcf somatic.pm.vcf --out somatic.pm --recode --recode-INFO-all

**SNP Eff**

java -Xmx4g -jar snpEff.jar -v hg19kg [somatic.pm](http://somatic.pm).recode.vcf -s somatic.pm.recode.vcf.html > somatic.pm.recode.ann.vcf

**SNP Sift**

java -Xmx4g -jar SnpSiftt.jar Annotate hapmap.\_3.3.b37.vcf [somatic.pm](http://somatic.pm).recode.ann.vcf > [somatic.pm](http://somatic.pm).recode.ann.hapmap.vcf

java -Xmx4g -jar SnpSiftt.jar Annotate clinvar\_Pathogenic.vcf >  [somatic.pm](http://somatic.pm).recode.ann.hapmap.vcf > [somatic.pm](http://somatic.pm).recode.ann.pathogenic.vcf

**FILTERING**

java -Xmx4g -jar SnpSift.jar filter "((exists CLNSIG) || ((ANN[ANY].IMPACT = 'HIGH') || (ANN[ANY].IMPACT = 'MODERATE') || ((ANN[ANY].IMPACT = 'LOW') && (ANN[ANY].EFFECT has 'splice\_region\_variant')) || ((ANN[ANY].IMPACT = 'MODIFIER') && ((ANN[ANY].EFFECT has '5\_prime\_UTR\_variant') || (ANN[ANY].EFFECT has '3\_prime\_UTR\_variant') || (ANN[ANY].EFFECT has 'regulatory\_region\_variant') || (ANN[ANY].EFFECT has 'TF\_binding\_site\_variant'))))) && (DP > 20) && (exists ID)" < somatic.pm.recode.ann.pathogenic.vcf > somatic.pm.recode.ann.pathogenic.SUPER\_FILTER.vcf

# Somatic copy number variant calling

samtools mpileup -q 1 -f human\_g1k\_v37.fasta  ControlGOD.bam TumorGOD.bam | java -jar VarScan.v2.3.9.jar copynumber --output-file SCNA --mpileup 1

java -jar VarScan.v2.3.9.jar copyCaller SCNA.copynumber --output-file SCNA.copynumber.called

**THE FILE “SCNA.copynumber.called” CAN BE USED IN THE CSB.R FILE**

# Purity and ploidy estimation (CLONET)

grep -E "(^#|0/1)" TumorGOD.bcf.filtered.recode.vcf > Tumor.het.vcf

grep -E "(^#|0/1)" ControlGOD.bcf.filtered.recode.vcf > Control.het.vcf

java -jar GenomeAnalysisTK.jar -T ASEReadCounter -R human\_g1k\_v37.fasta -o Tumor.csv -I TumorGOD.bam -sites Tumor.het.vcf -U ALLOW\_N\_CIGAR\_READS -minDepth 20 --minMappingQuality 20 --minBaseQuality 20

java -jar VarScan.v2.3.9.jar somatic ControlGOD.pileup TumorGOD.pileup --output-snp somatic.TPES.pm --output-indel somatic.TPES.indel

**THIS WILL GENERATE “somatic.TPSE.pm”, THEN YOU CAN USE CLONET.R SCRIPT**

# Ancestry Analysis

**SIMPLY RUN “EthSeq.R”**