# README

**Metrics by Sample**

* Apply PASS and FAIL filters based on hard-filters recommendations from GATK:

module load bcftools

for filepath in WES\_vcfs/\*.vcf.gz

do

filename=$(basename $filepath)

bcftools filter -s FAIL -i '(TYPE="snp" & INFO/QD>=2 & INFO/FS<=60 & INFO/MQ>=40 & INFO/SOR<=3 & (INFO/MQRankSum="." | INFO/MQRankSum>=-12.4) & (INFO/ReadPosRankSum="." | INFO/ReadPosRankSum>=-8.0)) | (TYPE~"indel" & INFO/QD>=2 & INFO/FS<=200 & INFO/SOR<=10 & (INFO/ReadPosRankSum="." | INFO/ReadPosRankSum>=-20))' $filepath -Oz -o WES\_filtered/${filename}

done

for filepath in ~/projects/def-vmooser/shared/exomeplus\_pilot/illumina\_novaseq/A00266\_0379/A00266\_0379\_2/\*.vcf.gz

do

filename=$(basename $filepath)

bcftools filter -s FAIL -i '(TYPE="snp" & INFO/QD>=2 & INFO/FS<=60 & INFO/MQ>=40 & INFO/SOR<=3 & (INFO/MQRankSum="." | INFO/MQRankSum>=-12.4) & (INFO/ReadPosRankSum="." | INFO/ReadPosRankSum>=-8.0)) | (TYPE~"indel" & INFO/QD>=2 & INFO/FS<=200 & INFO/SOR<=10 & (INFO/ReadPosRankSum="." | INFO/ReadPosRankSum>=-20))' $filepath -Oz -o A00266\_0379\_2\_filtered/${filename}

done

* run\_happy\_WGS.sh & run\_ha,ppy\_WES.sh
* happy\_metrics.R on WGS and WES .csv output directories
  + Creates ggplots of metrics recall & precision for PASS & FAIL
* Take only PASS

for filepath in WES\_filtered/\*.vcf.gz

do

filename=$(basename $filepath)

bcftools view -i "%FILTER!='FAIL'" $filepath -o WES\_filtered\_PASS/$filename

done

for filepath in WGS\_filtered/\*.vcf.gz

do

filename=$(basename $filepath)

bcftools view -i "%FILTER!='FAIL'" $filepath -o WGS\_filtered\_PASS/$filename

done

**Generate Flanking**

for i in $(seq 1 $END);

do

awk -v s=50 '{print $1, $2-s, $2}' CDS\_chrom/$i > CDS\_flank\_by\_chrom/$i

awk -v s=50 '{print $1, $3, $3+s}' CDS\_chrom/$i >> CDS\_flank\_by\_chrom/$i

tr -s ' \t' '\t\*' < CDS\_flank\_by\_chrom/$i > CDS\_tab/$i

sort -V CDS\_tab/$i > CDS\_flank\_final/$i

done

awk -F'\t' 'BEGIN{SUM=0}{ SUM+=$3-$2 }END{print SUM}' 1

cat X | awk '{for(i=$2;i<=$3;i++) print i}' | sort | uniq | wc -l

**Metrics by Depth Bins**

* vcf\_assign\_depth\_bins.py to assign depth bins

for filepath in ~/projects/def-vmooser/clarewei/happy/$protocol/WES\_filtered/\*.vcf.gz

do

filename=$(basename $filepath)

filename=("${filename%.hc.vcf.gz}")

python vcf\_assign\_depth\_bins.py -i $filepath -o ~/projects/def-vmooser/clarewei/metrics\_by\_depth\_bins/$protocol/WES\_depth\_bins/${filename}.vcf.gz -b 0 10 20 40 60 80 100 200

done

for filepath in ~/projects/def-vmooser/clarewei/happy/$protocol/WGS\_filtered/\*\_30\_E.vcf.gz

do

filename=$(basename $filepath)

python vcf\_assign\_depth\_bins.py -i $filepath -o ~/projects/def-vmooser/clarewei/metrics\_by\_depth\_bins/$protocol/WGS\_exome\_depth\_bins/${filename} -b 0 10 20 40 60 80 100 200

done

for filepath in ~/projects/def-vmooser/clarewei/happy/$protocol/WGS\_filtered/\*.vcf.gz

do

filename=$(basename $filepath)

filename=("${filename%.hc.vcf.gz}")

python vcf\_assign\_depth\_bins.py -i $filepath -o ~/projects/def-vmooser/clarewei/metrics\_by\_depth\_bins/$protocol/WGS\_depth\_bins/${filename}.vcf.gz -b 0 1 5 10 20 30

done

* Aggregate with happy output (TP/FP/FN)

module load bcftools

module load tabix

for filepath in \*.vcf.gz

do

filename=$(basename $filepath)

tabix $filepath

bcftools annotate -a $filepath -c "DP\_BIN" ~/projects/def-vmooser/clarewei/happy/$protocol/happy\_WES/${filename}| gzip -c > ../WES\_final/${filename}

done

for filepath in \*.vcf.gz

do

filename=$(basename $filepath)

tabix $filepath

bcftools annotate -a $filepath -c "DP\_BIN" ~/projects/def-vmooser/clarewei/happy/illumina/happy\_WGS/${filename}| gzip -c > ../WGS\_exome\_final/${filename}

done

for filepath in \*.vcf.gz

do

filename=$(basename $filepath)

tabix $filepath

bcftools annotate -a $filepath -c "DP\_BIN" ~/projects/def-vmooser/clarewei/happy/$protocol/happy\_WGS/${filename}| gzip -c > ../WGS\_final/${filename}

done

* Take only PASS

for filepath in WES\_final/\*.vcf.gz

do

filename=$(basename $filepath)

bcftools view -i "%FILTER!='FAIL'" $filepath -o WES\_final\_PASS/$filename

done

for filepath in WGS\_final/\*.vcf.gz

do

filename=$(basename $filepath)

bcftools view -i "%FILTER!='FAIL'" $filepath -o WGS\_final\_PASS/$filename

done

* Make count tables

module load nixpkgs/16.09

module load gcc/8.3.0

module load r/4.0.0

for file in ../$protocol/WES\_final\_PASS/\*

do

Rscript depth\_bins\_count\_tables.R $file ../$protocol/WES\_count\_tables

done

for file in ../$protocol/WGS\_final\_PASS/\*

do

Rscript depth\_bins\_count\_tables.R $file ../$protocol/WGS\_count\_tables

done

**Analyses by Gene**

BED files

* GTF annotation of ‘basic’ genes
  + <https://www.gencodegenes.org/human/release_34lift37.html>
* Daniel’s gencode\_cds and gencode\_utr.py to extract CDS regions as BED files
* Remove ‘chr’ prefix
  + cat genes\_cds.bed | sed 's/^chr//' > genes\_cds\_new.bed
* Appends GC
  + python bed\_compute\_gc.py -b ~/scratch/CDS/genes\_cds\_new.bed -f /cvmfs/soft.mugqic/CentOS6/genomes/species/Homo\_sapiens.GRCh37/genome/Homo\_sapiens.GRCh37.fa -o ~/scratch/GC/genes\_cds\_gc.bed
* Split bed by chrom
  + awk '{print>"CDS\_chrom/"$1}' genes\_cds\_gc.bed

VCFs from BAM

* make\_bam\_vcfs.sh
* make\_bam\_vcfs\_by\_chrom.sh
  + Inputs: Folder of bam\_vcfs
  + Outputs: Split folders containing info on each chromosome by sample
  + Additional needs: subfolders in output directory by name of sample

for filepath in bam\_vcfs/\*.vcf.gz

do

filename=$(basename $filepath)

outputDir=( "${filename%.vcf.gz}" )

mkdir bam\_vcfs\_by\_chrom/$outputDir

done

* + Function:
    - Creates text files of each sample with CHROM, POS, DP as columns
    - Splits text files up by chromosomes
    - Splits each chromosome into 1 million lines

Intersect BED with VCFs

* Intersect with each $chr
* Additional requirements: Sample name directory, each with directory of chromosomes 1-22

for filepath in bam\_vcfs/\*.vcf.gz

do

filename=$(basename $filepath)

outputDir=( "${filename%.vcf.gz}" )

mkdir CDS/split\_output/$outputDir

mkdir UTR/split\_output/$outputDir

mkdir CDS/combined/$outputDir

mkdir UTR/combined/$outputDir

END=22

for i in $(seq 1 $END)

do

mkdir CDS/split\_output/$outputDir/$i

mkdir UTR/split\_output/$outputDir/$i

done

done

* Submit batch jobs by chromosome
* Note: max ~ 1000

cd /home/clarewei/scratch/by\_gene/illumina/round2/overlap\_outlog

chr=1

for file in /home/clarewei/scratch/by\_gene/illumina/round2/bam\_vcfs/\*.vcf.gz

do

filename=$(basename $file)

sample\_dir=( "${filename%.vcf.gz}" )

sbatch --mem-per-cpu 2G --time 1:00:00 --wrap="module load nixpkgs/16.09; module load gcc/8.3.0; module load r/4.0.0; \

for vcf in /home/clarewei/scratch/by\_gene/illumina/round2/bam\_vcfs\_by\_chrom/$sample\_dir/$chr/\*

do

Rscript /home/clarewei/projects/def-vmooser/clarewei/by\_gene/by\_gene/findOverlap.R $vcf /home/clarewei/projects/def-vmooser/clarewei/by\_gene/CDS\_bed/$chr ../CDS/split\_output/$sample\_dir/$chr; \

Rscript /home/clarewei/projects/def-vmooser/clarewei/by\_gene/by\_gene/findOverlap.R $vcf /home/clarewei/projects/def-vmooser/clarewei/by\_gene/UTR\_bed/$chr ../UTR/split\_output/$sample\_dir/$chr

done"

done

cd /home/clarewei/scratch/by\_gene/mgi/round2/overlap\_outlog

for file in /home/clarewei/scratch/by\_gene/mgi/round2/bam\_vcfs/\*.vcf.gz

do

filename=$(basename $file)

sample\_dir=( "${filename%.vcf.gz}" )

sbatch --mem-per-cpu 2G --time 6:00:00 --wrap="module load nixpkgs/16.09; module load gcc/8.3.0; module load r/4.0.0; \

for vcf in /home/clarewei/scratch/by\_gene/mgi/round2/bam\_vcfs\_by\_chrom/$sample\_dir/$chr/\*

do

Rscript /home/clarewei/projects/def-vmooser/clarewei/by\_gene/by\_gene/findOverlap.R $vcf /home/clarewei/projects/def-vmooser/clarewei/by\_gene/CDS\_bed/$chr ../CDS/split\_output/$sample\_dir/$chr; \

Rscript /home/clarewei/projects/def-vmooser/clarewei/by\_gene/by\_gene/findOverlap.R $vcf /home/clarewei/projects/def-vmooser/clarewei/by\_gene/UTR\_bed/$chr ../UTR/split\_output/$sample\_dir/$chr

done"

done

cd /home/clarewei/scratch/by\_gene/illumina/round2/overlap\_outlog

FILES=(/home/clarewei/scratch/by\_gene/illumina/round2/bam\_vcfs/\*.vcf.gz)

chr=4

for file in ${FILES[@]:0:24}

do

filename=$(basename $file)

sample\_dir=( "${filename%.vcf.gz}" )

for vcf in /home/clarewei/scratch/by\_gene/illumina/round2/bam\_vcfs\_by\_chrom/$sample\_dir/$chr/\*

do

sbatch --mem-per-cpu 2G --wrap="module load nixpkgs/16.09; module load gcc/8.3.0; module load r/4.0.0; \

Rscript /home/clarewei/projects/def-vmooser/clarewei/by\_gene/by\_gene/findOverlap.R $vcf /home/clarewei/projects/def-vmooser/clarewei/by\_gene/CDS\_bed/$chr ../CDS/split\_output/$sample\_dir/$chr; \

Rscript /home/clarewei/projects/def-vmooser/clarewei/by\_gene/by\_gene/findOverlap.R $vcf /home/clarewei/projects/def-vmooser/clarewei/by\_gene/UTR\_bed/$chr ../UTR/split\_output/$sample\_dir/$chr"

done; done

cd /home/clarewei/scratch/by\_gene/mgi/round2/overlap\_outlog

FILES=(/home/clarewei/scratch/by\_gene/mgi/round2/bam\_vcfs/\*.vcf.gz)

for file in ${FILES[@]:0:24}

do

filename=$(basename $file)

sample\_dir=( "${filename%.vcf.gz}" )

for vcf in /home/clarewei/scratch/by\_gene/mgi/round2/bam\_vcfs\_by\_chrom/$sample\_dir/$chr/\*

do

sbatch --mem-per-cpu 2G --wrap="module load nixpkgs/16.09; module load gcc/8.3.0; module load r/4.0.0; \

Rscript /home/clarewei/projects/def-vmooser/clarewei/by\_gene/by\_gene/findOverlap.R $vcf /home/clarewei/projects/def-vmooser/clarewei/by\_gene/CDS\_bed/$chr ../CDS/split\_output/$sample\_dir/$chr; \

Rscript /home/clarewei/projects/def-vmooser/clarewei/by\_gene/by\_gene/findOverlap.R $vcf /home/clarewei/projects/def-vmooser/clarewei/by\_gene/UTR\_bed/$chr ../UTR/split\_output/$sample\_dir/$chr"

done; done

* Output: csvs of intersection by split chromosomes in respective folders

END=14619325

for i in $(seq 14618764 $END)

do scancel $i; done

Combine by Chromosome

* Input: directories containing all intersection of one chromosome
* Output: 1 csv file of that chromosome

cd /home/clarewei/scratch/GC/combine\_outlog

END=22

for chr in $(seq 1 $END)

do

for file in /home/clarewei/scratch/bam\_vcfs/\*.vcf.gz

do

filename=$(basename $file)

sample\_dir=( "${filename%.vcf.gz}" )

sbatch --mem-per-cpu 2G --wrap="module load nixpkgs/16.09; module load gcc/8.3.0; module load r/4.0.0; Rscript ../combine\_by\_chrom.R ~/scratch/GC/CDS\_output/$sample\_dir/$chr ~/scratch/GC/CDS\_output/$sample\_dir; Rscript ../combine\_by\_chrom.R ~/scratch/GC/UTR\_output/$sample\_dir/$chr ~/scratch/GC/UTR\_output/$sample\_dir"

done

done

* Copy csvs by chromosome to separate directory

for filepath in /home/clarewei/scratch/bam\_vcfs/\*.vcf.gz

do

filename=$(basename $filepath); outputDir=( "${filename%.vcf.gz}" )

cp CDS\_output/$outputDir/\*.csv CDS\_combined/$outputDir

cp UTR\_output/$outputDir/\*.csv UTR\_combined/$outputDir

done

Generating Results

* INPUT\_DIR — CDS\_combined or UTR\_combined
* Outputs: boxplots by average, median, stats\_summary, and coverage~gc content

Rscript plot\_by\_gene <INPUT\_DIR> <OUTPUT\_DIR>

**Indel Lengths**

* Create complete PASS vcfs

for filepath in WGS\_filtered/\*.vcf.gz

do

filename=$(basename $filepath)

bcftools view -i "%FILTER!='FAIL'" $filepath -o WGS\_filtered\_PASS/$filename

done

for filepath in WES\_filtered/\*.vcf.gz

do

filename=$(basename $filepath)

bcftools view -i "%FILTER!='FAIL'" $filepath -o WES\_filtered\_PASS/$filename

done

* Find intersection of WGS with exome target bed

for filepath in ~/projects/def-vmooser/clarewei/happy/illumina/WGS\_filtered\_PASS/\*.vcf.gz

do

filename=$(basename $filepath)

bedtools intersect -a $filepath -b /lustre03/project/6050814/shared/benchmark/bin/SureSelectHumanAllExonV6\_UTR.target.bed -header | gzip -c > /home/clarewei/projects/def-vmooser/clarewei/indel\_lengths/illumina/WGS\_intersect/$filename

bedtools subtract -a $filepath -b /lustre03/project/6050814/shared/benchmark/bin/SureSelectHumanAllExonV6\_UTR.target.bed -header | gzip -c > /home/clarewei/projects/def-vmooser/clarewei/indel\_lengths/illumina/WGS\_subtract/$filename

done

* make\_indel\_count\_tables.R to assign indel length bins

for filepath in ~/projects/def-vmooser/clarewei/happy/illumina/WES\_filtered\_PASS/\*.vcf.gz

do

Rscript make\_indel\_count\_tables.R $filepath ../illumina/WES\_count\_tables

done

for filepath in ~/projects/def-vmooser/clarewei/happy/illumina/WGS\_filtered/\*.vcf.gz

do

Rscript make\_indel\_count\_tables.R $filepath ../illumina/WGS\_count\_tables

done

* make\_indel\_count\_tables on intersecting and subtracted only

for filepath in ~/projects/def-vmooser/clarewei/indel\_lengths/illumina/WGS\_subtract/\*.vcf.gz

do

filename=$(basename $filepath)

Rscript make\_indel\_count\_tables.R $filepath ../illumina/WGS\_subtract\_count\_tables

Rscript make\_indel\_count\_tables.R ~/projects/def-vmooser/clarewei/indel\_lengths/illumina/WGS\_intersect/$filename ../illumina/WGS\_intersect\_count\_tables

done