Supplementary Information

DNN-Boost: Somatic Mutation Identification of Tumor-Only Whole-Exome Sequencing Data Using Deep Neural Network and XGBoost

Firda Aminy Maruf[1], Rian Pratama[1], Giltae Song[1]
[1]School of Computer Science and Engineering, Pusan National University, 63 Busandaehak-Ro, Busan 46241, Republic of Korea

Supplementary 1. Materials & Methods

WES Alignment with Bowtie2

1) Map reads against reference genome:

```
bowtie2 --end-to-end --very-fast --rg-id [ID FOR THE PAIRED-END READS] -x GRCH38 -q -1 [INPUT FASTQ FILE PAIR 1] -2 [INPUT FASTQ FILE PAIR 2] | samtools view - -Sb -h -t GCA_000001405.15_GRCh38_no_alt_analysis_set.fna.fai -o [OUTPUT BAM FILE]
```

2) Sort the output BAM file with SAMTOOLS:

```
samtools sort [INPUT BAM FILE] -o [OUTPUT SORTED BAM FILE] -m 8000000000
```

3) Remove PCR duplicates with SAMTOOLS:

```
samtools rmdup [INPUT SORTED BAM FILE] [OUTPUT SORTED DEDUPLICATED BAM FILE]
```

Variant Calling

Mutect2

1) Run Mutect2 w/matched normal for the benchmark set

```
gatk --java-options "-Xmx8g" Mutect2 -R GCA_000001405.15_GRCh38_no_alt_analysis_set.fna -I [INPUT
SORTED DEDUPLICATED TUMOR BAM FILE] -I [INPUT SORTED
DEDUPLICATED NORMAL BAM FILE] -tumor [ID TUMOR BAM FILE] -normal
[ID NORMAL BAM FILE] -pon [PON VCF.gz FILE] --germline-resource
somatic-hg38_af-only-gnomad.hg38.vcf --af-of-alleles-not-in-
resource 0.0000025 -L
Homo_sapiens_assembly38_exome.targets.interval_list -O [OUTPUT
VCF FILE]
```

2) Run FilterMutectCalls to filter somatic variants, germline variants, and artifacts in the Mutect2 VCF callset

```
gatk FilterMutectCalls -R
GCA_000001405.15_GRCh38_no_alt_analysis_set.fna -V [INPUT
MUTECT2 UNFILTERED VCF] -O [OUTPUT MUTECT2 FILTERED VCF]
```

3) Filter out the indels from the Mutect2 filtered VCFs callset

```
gatk SelectVariants -V [INPUT MUTECT2 FILTERED VCF] -select-type SNP -O
[OUTPUT MUTECT2 SNP-ONLY VCF]
```

4) Annotate each of the SNP-only VCFs with ANNOVAR to acquire the functional prediction features:

```
perl table_annovar.pl [INPUT MUTECT2 FILTERED VCF] humandb/ -
buildver hg38 -out [OUTPUT ANNOTATED VCF] --remove --protocol
refGene,exac03,avsnp150,dbnsfp33a,gnomad_exome,cosmic92_coding
,clinvar_20210123 --operation gx,f,f,f,f,f,f -nastring . -
vcfinput -polish -xref example/gene_fullxref.txt
```

HaplotypeCaller

1) Run the HaplotypeCaller on each tumor and normal samples BAM files to create single-sample gVCFs, with the option --emitRefConfidence GVCF, and using the .g.vcf extension for the output file.

```
gatk --java-options "-Xmx4g" HaplotypeCaller -R
GCA_000001405.15_GRCh38_no_alt_analysis_set.fna -I [INPUT
SORTED DEDUPLICATED TUMOR BAM FILE] -O [OUTPUT .g.vcf] -A
StrandBiasBySample -ERC GVCF
```

2) Aggregate the multiple GVCF files:

```
gatk --java-options "-Xmx96g -Xms96g" CombineGVCFs -R
GCA_000001405.15_GRCh38_no_alt_analysis_set.fna -V
[INPUT .g.vcf] -V [INPUT .g.vcf] -V [INPUT .g.vcf] -V
[INPUT .g.vcf] -O [OUTPUT FILE COHORT .g.vcf]
```

3) Joint genotyping

```
gatk --java-options "-Xmx4g" GenotypeGVCFs -R
GCA_000001405.15_GRCh38_no_alt_analysis_set.fna -V [INPUT FILE
COHORT .g.vcf] -O [OUTPUT FINAL COHORT VCF]
```

4) Subset to SNPs-only callset with SelectVariants

```
gatk SelectVariants -V [INPUT FINAL COHORT VCF] -select-type SNP
-O [OUTPUT SNP-ONLY VCF]
```

5) Hard-filtering variant

```
gatk VariantFiltration -V [INPUT SNP-ONLY VCF] -filter "QD <
2.0" --filter-name "QD2" -filter "QUAL < 30.0" --filter-name
"QUAL30" -filter "FS > 60.0" --filter-name "FS60" -filter "MQ <
40.0" --filter-name "MQ40" -filter "MQRankSum < -12.5" --filter-
name "MQRankSum-12.5" -O [OUTPUT FILTERED SNP-ONLY VCF]
```

BCFtools

1) Create a list of bams to use:

```
ls *.bam > [OUTPUT BAMLIST .txt]
```

2) Pile the multiple samples, call variants according to the targeted regions, and pipe it to bcftools to create a VCF file:

```
bcftools mpileup -d 250 -R [INPUT TARGETED REGIONS BED FILE] -B
-Ou   -f   GCA_000001405.15_GRCh38_no_alt_analysis_set.fna   -b
[INPUT BAMLIST .txt] | bcftools call -mv -O v -o [OUTPUT VCF]
```

3) Filter query for the variants calling results:

```
bcftools  filter  -sLowQual  -g3  -G10  -e'%QUAL<10  ||  (RPB<0.1
&&  %QUAL<15)  ||  (AC<2  &&  %QUAL<15)'  [INPUT  VCF]  >  [OUTPUT
FILTERED VCF]
```