Variant calling for chr19 using GATK haplotypecaller

Neil Zhang

1. QC for the bam file

1.1 Pre-process the bed file

Qualimap requires the bed file to have 6 columns while the provided one only has 3. Add three extra columns:

```
1. awk 'BEGIN{OFS="\t"}{print $1 ,$2,$3,".",".",".","."}' target_regions.bed >
   target_regions_new.bed
```

1.2 Quality control for the bam file using Qualimap

```
1. #-os: add analysis for outside regions
2. ~/bioinfo_tools/qualimap_v2.2.1/qualimap bamqc --java-mem-size=20G -gff
   /home/neil/sophiag_vc/raw_data/target_regions_new.bed -os -bam
   /home/neil/sophiag_vc/raw_data/chr19.bam -outdir
   /home/neil/sophiag_vc/output/qualimap
```

From the result of Qualimap, there are 1.8 million reads in the bam file. 83.95% are mapped to the target regions, 15.67% are mapped to the outside regions. The mean mapping qualities for both inside and outside reads are bigger than 58.

First, I checked the read coverage for the target regions, the mean coverage is 198X. From the coverage histogram in Figure 1, we can see that the distribution is similar to a normal distribution, most locations have coverage more than 60X. The mean GC-content for the target regions is 62.03%, which is little higher than the human genome average.

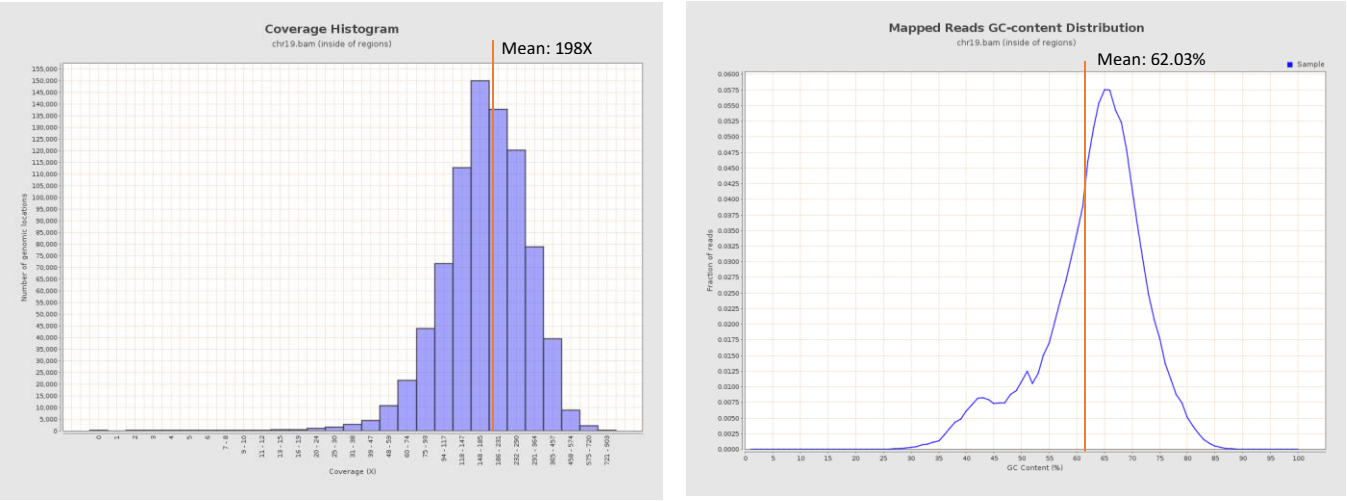


Figure 1. Left: Coverage histogram of locations inside target regions. Right: Distributions of GC-content for reads inside target regions.

Next, I checked the read coverage for the outside regions. We can see that most locations outside the target region are not converged by any reads, which indicates the target enrichment is very specific.

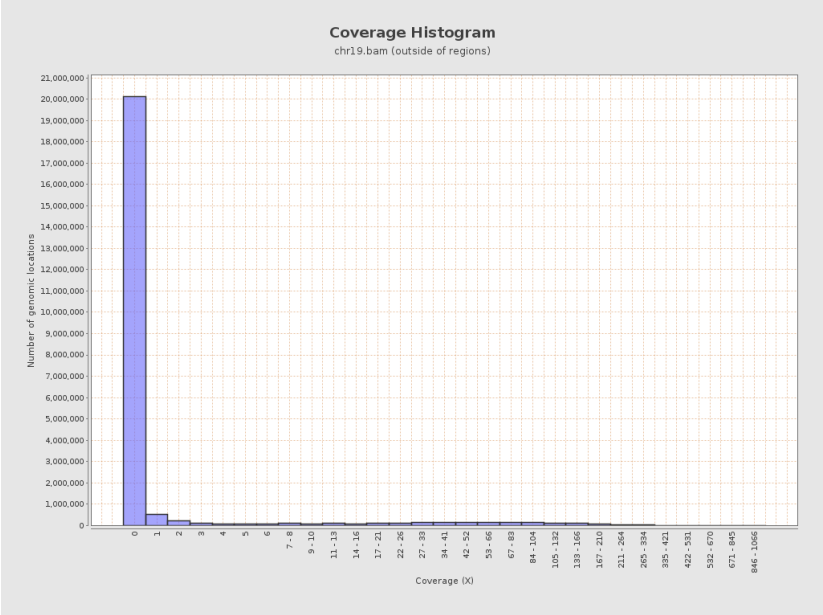


Figure 2. Coverage histogram of locations outside target regions.

In summary, the analysis suggests that most reads are specific to target regions and mapped to the genome with high confidence. Locations within the target regions have high sequencing depth. The quality of the bam file is good for the downstream variant calling.

2. Variant calling through GATK

2.1 Check the bam file

```
1. samtools view -h chr19.bam | head -100
```

The bam file is already sorted according to the mapping location.

2.2 Remove duplicates in the BAM file and build index:

```
1. /home/neil/bioinfo_tools/gatk-4.1.9.0/gatk --java-options "-Xmx20G"
   MarkDuplicates \
2.   -I chr19.bam --REMOVE_DUPLICATES true  -O chr19_marked.bam -M
   chr19.metrics
```

I got an error message "Read name A00129:298:H7JGHDSXX:1:1101:26838:30029 No real operator (M|I|D|N) in CIGAR". I searched for this read and found that its CIGAR is "133S". Then I tried to search for all reads whose CIGAR only contains "S" (soft clip):

```
samtools view chr19.bam | awk '$6~/^[0-9]*S$/'
```

I found that there are 195 reads and all of them are mapped to 4511282. There may be something wrong at this location during genome mapping. I filtered them out, saved the SAM file and re-ran MarkDuplicates.

```
samtools view chr19.bam | awk '$6!~/^[0-9]*S$/' > chr19_f.sam
```

2.3 Build index for the BAM

```
samtools index -@24 -m 20G -b chr19_marked.bam
```

2.4 Download the reference genome and the common variants in human genome

From the head of BAM and vcf file, we see that the reference genome is HumanG1Kv37 (b37,hg19)

```
# reference genome
wget
http://ftp.1000genomes.ebi.ac.uk/vol1/ftp/technical/reference/human_g1k_v37.fasta.gz
# common variants in human genome
wget
https://data.broadinstitute.org/snowman/hg19/variant_calling/vqsr_resources/WGS/v2/1000G_phase1.snps.high_confidence.b37.vcf.gz
wget
https://data.broadinstitute.org/snowman/hg19/variant_calling/vqsr_resources/WGS/v2/1000G_phase1.snps.high_confidence.b37.vcf.gz.tbi
wget
https://data.broadinstitute.org/snowman/hg19/variant_calling/vqsr_resources/WGS/v2/Mills_and_1000G_gold_standard.indels.b37.vcf.gz
wget
https://data.broadinstitute.org/snowman/hg19/variant_calling/vqsr_resources/WGS/v2/Mills_and_1000G_gold_standard.indels.b37.vcf.gz.tbi
```

Build index and dictionary for the reference genome:

```
samtools faidx human_g1k_v37.fasta.gz
/home/neil/bioinfo_tools/gatk-4.1.9.0/gatk --java-options "-Xmx20G"
CreateSequenceDictionary -R human_g1k_v37.fasta.gz
```

2.5 Base quality score recalibration using BQSR

The original bam file doesn't have the read group information, which is needed for BQSR. I added the read group information to the bam file:

```
java -jar /home/neil/bioinfo_tools/gatk-4.1.9.0/picard.jar
AddOrReplaceReadGroups \
  -I chr19_marked.bam -O chr19_markedrg.bam  -RGID 1  -RGLB lib1  -RGPL
UNKNOWN -RGPU unit 1 -RGSM 1
```

Generate recalibration table:

```
1. /home/neil/bioinfo_tools/gatk-4.1.9.0/gatk --java-options "-Xmx20G"
   BaseRecalibrator \
2.   -R ~/bioinfo_tools/Reference/human/human_g1k_v37.fasta.gz \
3.   -I chr19_markedrg.bam \
4.   --known-sites
   /home/neil/bioinfo_tools/Reference/human/1000G_phase1.snps.high_confide
   nce.b37.vcf.gz \
5.   --known-sites
   /home/neil/bioinfo_tools/Reference/human/Mills_and_1000G_gold_standard.
   indels.b37.vcf.gz\
6.   -O chr19_recal.table
```

Perform base quality score recalibration:

```
1. /home/neil/bioinfo_tools/gatk-4.1.9.0/gatk --java-options "-Xmx20G"
   ApplyBQSR \
2.   -R ~/bioinfo_tools/Reference/human/human_g1k_v37.fasta.gz \
3.   -I chr19_markedrg.bam \
4.   -bqsr chr19_recal.table \
5.   -O chr19_bqsr.bam
```

2.6 Germline mutations detection for target regions using HaplotypeCaller

Generate the intermediate g.vcf:

```
1. /home/neil/bioinfo_tools/gatk-4.1.9.0/gatk --java-options "-Xmx20G"
   HaplotypeCaller \
2.  -R ~/bioinfo_tools/Reference/human/human_g1k_v37.fasta.gz \
3.  -L target_regions.bed \
4.  --emit-ref-confidence GVCF \
5.  -I chr19_bqsr.bam \
6.  -O chr19_tr.g.vcf
```

Generate the vcf file:

```
1.  /home/neil/bioinfo_tools/gatk-4.1.9.0/gatk --java-options "-Xmx20G"
   GenotypeGVCFs \
2.  -R ~/bioinfo_tools/Reference/human/human_g1k_v37.fasta.gz  -V
   chr19_tr.g.vcf  -O chr19_tr.vcf
```

2.7 Separate SNPs and Indels to apply different filters

```
1. /home/neil/bioinfo_tools/gatk-4.1.9.0/gatk SelectVariants \
2.     -V chr19_tr.vcf  -select-type SNP  -O chr19_tr_snp.vcf
3. /home/neil/bioinfo_tools/gatk-4.1.9.0/gatk SelectVariants \
4.     -V chr19_tr.vcf  -select-type INDEL -O chr19_tr_indel.vcf
5. bcftools index chr19_tr_indel.vcf
6. bcftools index chr19_tr_snp.vcf
```

2.8 Filter out low quality variants through hard filtering

In the last step, I filtered out the low-quality variants. Since I only have limited number of samples, it is not appropriate to use VQSR. I manually selected the threshold for the variant features and performed hard filtering.

7 features for the variants are used for filtering low quality variants: Quality (QUAL), QualByDepth (QD), StrandOddsRatio (SOR), FisherStrand (FS), RMSMappingQuality (MQ), MappingQualityRankSumTest (MQRankSum), ReadPosRankSumTest (ReadPosRankSum). The key is to choose the threshold for each feature.

For example, to choose thresholds for QD and SOR. I first extracted feature values of all SNPs.

```
1. /home/neil/bioinfo_tools/gatk-4.1.9.0/gatk VariantsToTable \
2.       -V chr19_tr_snp.vcf  -O snp.table \
3.       -F CHROM -F POS -F QUAL -F QD -F SOR -F FS -F MQRankSum -F
   ReadPosRankSum
```

Then, I ploted feature histograms in R:

```
1. ggplot(snp_feature, aes(x=QD)) +
2.   geom_histogram(binwidth=2)+xlab('QD value for each variant site')+
3.   theme(axis.text=element_text(size=12))
4. ggplot(snp_feature, aes(x=SOR)) +
5.   geom_histogram(binwidth=0.2)+xlab('SOR value for each variant
   site')+theme(axis.text=element_text(size=12))
```

QD is the ratio between variants quality and sequencing depth, higher QD indicates higher confidence. There are two peaks in the histogram, the first peak corresponds to the heterozygous variants, the second peak corresponds to the homozygous variants. I removed the left tail at OD 6. SQR measures strand bias, higher value indicates bigger bias. I removed the right tail at SOR 3.

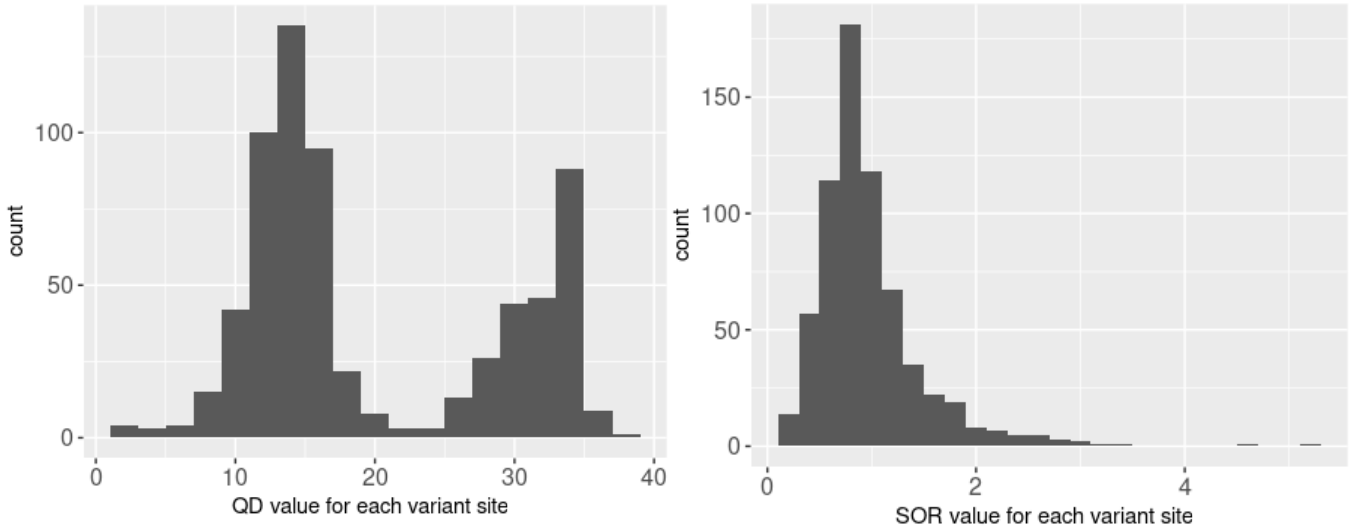


Figure 3. QD, SOR distributions for all SNPs in the raw result

Filter out low quality SNPs:

```
/home/neil/bioinfo_tools/gatk-4.1.9.0/gatk VariantFiltration \
    -V chr19_tr_snp.vcf \
    -filter "QD < 6.0" --filter-name "QD6.0" \
    -filter "QUAL < 200.0" --filter-name "QUAL200.0" \
    -filter "SOR > 3.0" --filter-name "SOR3" \
    -filter "FS > 40.0" --filter-name "FS40" \
    -filter "MQ < 40.0" --filter-name "MQ40" \
    -filter "MQRankSum < -10.0" --filter-name "MQRankSum-10" \
    -filter "ReadPosRankSum < -3.0" --filter-name "ReadPosRankSum-3" \
     -O chr19_tr_snp_vsqr.vcf
 #Select the passed variants
 cat chr19_tr_snp_vsqr.vcf | grep "#" > chr19_tr_snp_vsqr_f.vcf
 cat chr19_tr_snp_vsqr.vcf | grep "PASS" >> chr19_tr_snp_vsqr_f.vcf
```

Filter out low quality Indels:

```
/home/neil/bioinfo_tools/gatk-4.1.9.0/gatk VariantFiltration \
    -V chr19_tr_indel.vcf \
    -filter "QD < 8.5" --filter-name "QD8.5" \
    -filter "QUAL < 950.0" --filter-name "QUAL950" \
    -filter "SOR > 2.5" --filter-name "SOR2p5" \
    -filter "FS > 6.0" --filter-name "FS6" \
    -filter "ReadPosRankSum < -2.0" --filter-name "ReadPosRankSum-2" \
    -filter "MQRankSum < -0.5" --filter-name "MQRankSum0" \
    -O chr19_tr_indel_vsqr.vcf
 cat chr19_tr_indel_vsqr.vcf | grep "#" > chr19_tr_indel_vsqr_f.vcf
 cat chr19_tr_indel_vsqr.vcf | grep "PASS" >> chr19_tr_indel_vsqr_f.vcf
```

Merge the SNPs and Indels:

```
/home/neil/bioinfo_tools/gatk-4.1.9.0/gatk MergeVcfs \
    -I chr19_tr_indel_vsqr_f.vcf \
    -I chr19_tr_snp_vsqr_f.vcf \
    -O chr19_tr_vsqr_f.vcf
```

3. Variant calling performance

To analyze the performance of GATK HaplotypeCaller on our sample, I selected the shared variants between the result and the ground truth as well as the unique variants in each file using bcftools.

```
#the shared variants (n=2) between the filtered result and the ground truth
bcftools isec -p dir -n=2  -c both chr19_tr_vsqr_f.vcf.gz ground_truth.vcf.gz
#the unique variants (n=1) in the filtered result and the ground truth
bcftools isec -p dir -n=1  -c both chr19_tr_vsqr_f.vcf.gz ground_truth.vcf.gz
```

Calculate the performance confusion matrix:

```
#Total length of the target region
cat target_regions.bed | awk -F'\t' 'BEGIN{SUM=0}{ SUM+=$3-$2 }END{print SUM}'
#The number of unique variants in filtered GATK result
cat gatk_unique.vcf | grep -v "#" | wc -l
#The number of unique variants in ground truth
cat gt_unique.vcf | grep -v "#" | wc -l
#The number of shared variants
cat shared.vcf | grep -v "#" | wc -l
```

Total length of the region: 808887

True positives (the variants in both the result and the ground truth): 539

False positives (the variants only in the result): 127

False negatives (the variants only in the ground truth): 9

True negatives: 808887-539-127-9=808212

Sensitivity=539/(539+9)=98.36%

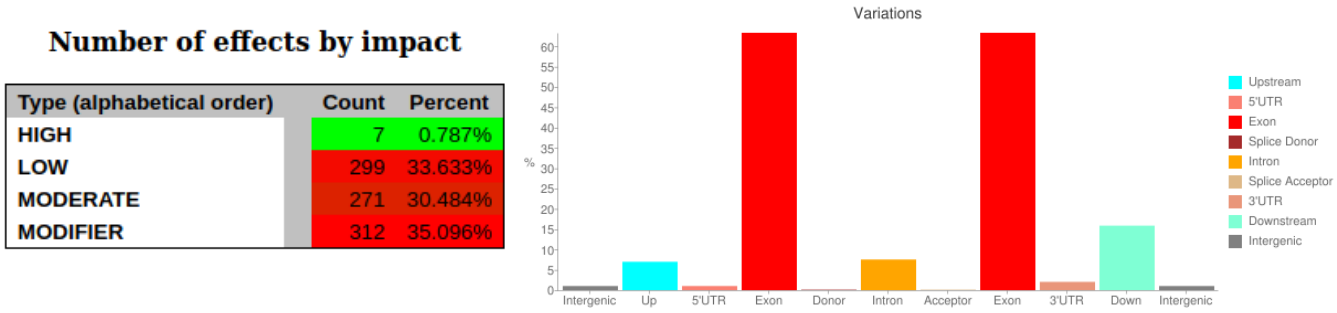
Precision=539/(539+127)=80.93%

Specificity=808212/(808212+127)=99.98%

4. Variants annotation using SnpEff

```
java -jar ~/bioinfo_tools/snpEff/snpEff.jar GRCh37.p13.RefSeq NeilZhang_chr19vcf_final.vcf -canon > chr19_snpEff.vcf
```

Finally, SnpEff was used to annotate the variants. SnpEff maps the variants to genes and predicts the effect of each variant. It generates a table with all the affected genes and some statistics for all the variants. From Figure 4, we can see that 7 variants are predicted to have a significant effect on gene functions. As expected, most variants are in the exon regions.



Number of effects by impact

Type (alphabetical order)	Count	Percent
HIGH	7	0.787%
LOW	299	33.633%
MODERATE	271	30.484%
MODIFIER	312	35.096%

Figure 4. Outputs from SnpEff. Left, the number of effects by impact on gene function. Since one variant can affect several transcripts generated by alternative splicing. The total number of effects is bigger than the total number of variants. Right, the gene regions that variants are located in.

Questions:

• Can you infer the type of sample (germline, somatic)?

The allele fraction for germline mutations should be 100% (homozygous) or 50% (heterozygous). I extracted allele frequency (AF) from the final vcf file. I found that the allele frequency for all variants is either 100% or 50%. This suggests that the sample is a germline sample.

```
1. /home/neil/bioinfo_tools/gatk-4.1.9.0/gatk VariantsToTable -V
   NeilZhang_chr19vcf_final.vcf  -O af.table -F CHROM -F POS -F AF
```

To identify somatic mutations, it will be better to have paired tumor and normal tissues.

• Is the variant calling strategy you chose appropriate for the type of sample?

From the analysis above, we can see that the variant calling has high sensitivity, precision and specificity. Therefore, the variant calling and filtering strategy are appropriate for this type of sample. We can also further fine tune the feature filtering thresholds to change the balance between sensitivity, precision and specificity for different purpose.

• What are (according to you) the reasons behind the FPs and the FNs you might observe?

FPs can be caused by sequencing errors or realignment mistakes (Indels).

FNs can be caused by low sequencing depth. As a result, variants are filtered out.