SNP and INDEL Analysis

Leila Taher

WS 2022/2023

1 Variant Calling and Annotation

1.1 Data set

This tutorial will take you through the process of calling variants with GATK using cattle whole genome sequencing data.

There are two BAM files containing the read mappings to an approximately 4 Mb-long region on chromosome 4, centered on the *Leptin* gene. You can view the region here. Each BAM file corresponds to a different cow.

The reads were mapped to the domestic cow (*Bos taurus*) genome (UMD 3.1 assembly) with the Burrows-Wheeler Aligner (BWA), a software package for mapping low-divergent sequences against a large reference genome, such as the human genome.

1.2 Tools

- SAMtools (http://www.htslib.org/doc/samtools.html)
- Picard (https://broadinstitute.github.io/picard/) (Installation in conda environment recommended)
- GATK (https://software.broadinstitute.org/gatk/) (Installation in conda environment recommended)
- VCFtools (https://vcftools.github.io/index.html)
- VEP (www.ensembl.org/vep) (Please, check end of PDF for help)

Download and install the tools if necessary.

1.3 Variant calling with GATK

1.3.1 1. Preparing the reference genome for use with GATK

GATK requires two partner files for the FASTA file with the reference genome used for the mapping: - a FASTA index file (.fai), and - a dictionary file (.dict).

- Download the FASTA file with the reference genome from Ensembl.

- Examine the BAM file (via samtools view). What is the name of the sequence to which the reads have been mapped? Uncompress the FASTA file and modify the name of the sequence(s) in it to match the BAM files.

- Use SAMtools to generate an index of the FASTA file:

```
samtools faidx Bos_taurus.UMD3.1.dna.chromosome.4.fa
```

- Now use Picard to generate the dictionary:

```
java -jar picard CreateSequenceDictionary \
 R=Bos_taurus.UMD3.1.dna.chromosome.4.fa \
 O=Bos_taurus.UMD3.1.dna.chromosome.4.dict
```

1.3.2 2. Sorting and indexing the BAM files

Sort and index your BAM files using SAMtools.

1.3.3 3. Mark duplicates

PCR duplicates can cause problems when calling variants.

- Use Picard to flag the reads that are potential duplicates:

```
java -jar picard MarkDuplicates \
 I=bam_file_sorted.bam \
 O=bam_file_sorted_dupMarked.bam \
 M=bam_file_sorted_dup_metrics.txt
```

Use SAMtools to sort and index the resulting BAM file.
What proportion of the reads are duplicates?

1.3.4 4. Base Quality Score Recalibration (BQSR)

BQSR adjusts the base quality scores so that they more accurately represent the probability of having called the wrong base. BQSR requires a set of known variants.

The following two steps were already performed for you. You received the output file with the exercise download and therefore you do not have to perform these two steps. They are just described here for completeness.

We will use the variants from this study, which could be downloaded as follows:

```
wget ftp://ftp.ebi.ac.uk/pub/databases/nextgen/bos/variants/population_sites/
UGBT.population_sites.UMD3_1.20140307.vcf.gz
```

The variants are in VCF format. Since we are only working with reads mapped to chromosome 4, the variants on this chromosome could be extracted using VCFtools:

```
vcftools --gzvcf UGBT.population_sites.UMD3_1.20140307.vcf.gz \
 --chr 4 --recode \
 --out UGBT.population_sites.UMD3_1.20140307.Chr4
```

Check the manual for the meaning of the four arguments. Finally, the name of the chromosome has been modified to match the BAM files. The file is called UGBT.population_sites.UMD3_1.20140307.Chr4.recode.vcf and is provided together with the BAM files.

Index the VCF file:

```
gatk IndexFeatureFile \
     --input UGBT.population_sites.UMD3_1.20140307.Chr4.recode.vcf
```

Use this file of known sites to compute the statistics for BQSR:

```
gatk BaseRecalibrator \
   -I bam_file_sorted_dupMarked.bam \
   -R Bos_taurus.UMD3.1.dna.chromosome.4.fa \
   --known-sites UGBT.population_sites.UMD3_1.20140307.Chr4.recode.vcf  \
   -O output_BaseRecalibrator.table
```

And apply the recalibration:

```
gatk ApplyBQSR \
   -I bam_file_sorted_dupMarked.bam \
   -R Bos_taurus.UMD3.1.dna.chromosome.4.fa \
   --bqsr-recal-file output_BaseRecalibrator.table \
   -O bam_file_sorted_dupMarked_recal.bam
```

1.3.5 5. Variant calling

Since the BAM files only contains reads mapped to a small region of the cow genome, we will restrict variant calling to this region. To do this, we first need to create a *target file*, defining the region of interest. Using a text editor, store the chromosome, start position, and end position in a file called “targetRegion.bed”:

Chr4 91249874 95266624

We will then use this targetRegion.bed with the –L flag to restrict calling to this region. If you wanted to call variants across the whole genome you would just leave this out:

```
gatk HaplotypeCaller \
 -R Bos_taurus.UMD3.1.dna.chromosome.4.fa \
 -L targetRegion.bed \
 -I bam_file_sorted_dupMarked_recal.bam \
 --dbsnp UGBT.population_sites.UMD3_1.20140307.chr4.recode.vcf \
 -o output_VCF_file
```

1.4 7. Variant annotation

VEP (variant effect predictor) is a commonly used tool to annotate genetic variants. To speed up annotation, you could run VEP specifying the location of a VEP local cache that has been installed on your computer. We will connect to the Ensembl servers to get this information.

```
vep --database \
 --species bos_taurus \
 --vcf -i variants.vcf \
 -o variants_VEP.vcf \
 --force_overwrite
```

Have a look at the html output file.

- How many missense variants were identified?
- Check the manual and work out how to add SIFT scores to the output vcf. You may need to delete the previous output file if using the same name. What proportion of the missense variants are predicted to be deleterious with higher confidence?

1.5 8. Variant filtering

VEP comes with a script that can be used to filter for specific types of variants. For example, it is possible to use the filter_vep.pl script to just extract the missense variants from the VEP annotated file.

```
filter_vep -I variants_VEP.vcf --format vcf -o missense_only.vcf \
 --filter "Consequence matches missense" --force_overwrite
```

- Get a count of the number of coding variants in a specific gene gene (hint take a look here).

1.6 9. Variant metrics

VCFtools is a program for manipulating VCF files. As well as getting various summary statistics it can also be used to convert VCF files to different formats. For example, to get the allele frequency of each variant:

```
vcftools -vcf variants_VEP.vcf --freq --out freqs
```

Examine the output files. To get counts of each type of change e.g. (A<->C) and the total number of transitions and transversions.

```
vcftools -vcf variants_VEP.vcf --TsTv-summary --out TsTv
```

Examine the output files.

The above two steps output many warnings but as long as the desired output files are received, they are not to worry about.

- Which base changes (e.g. A<->C, A<->G, etc) are observed most often in this data? Why might these be most common (hint: https://en.wikipedia.org/wiki/Deamination).
- How can you output a new VCF file with only the genotypes for the first sample?

2 VEP installation hints

We experienced, if your vep is not working immediately, these steps could help:

```
git clone https://github.com/Ensembl/ensembl-vep.git
cd ensembl-vep/
sudo apt-get install libdbi-perl
sudo apt install cpanminus
cpanm -S Bio::Root::Version
cpanm -S Try::Tiny
perl INSTALL.pl
sudo apt-get install libdbd-mysql-perl
```

And then run vep from the installed folder directly

3 Software versions used

samtools (standard, 1.10), picard (conda, 2.18.29), gatk (conda, 4.1.7.0), vcftools (conda, 0.1.16), vep (standard, 101.0)