Variant calling – sample-wise genotyping & lineage typing

Bharkbhoom Jamesia & Pakorn Aiewsakun

Pornchai Matangkasombut Center for Microbial Genomics

Department of Microbiology, Faculty of Science, Mahidol University

# Introduction

We now have a set of sequence reads that are aligned to the reference genome and stored as BAM files. What we will do next is to *‘genotype’* (or *‘call’*) your samples from their read mapping alignment files. There are several variant callers that can do this, such as BCFtools, iVar, and VarScan2.

In this practical, we will use *‘****G****enome* ***A****nalysis* ***T****ool****k****it’* (*‘GATK’*), specifically the program ‘*GATK HaplotypeCaller*’, to do per-sample variant calling, which will create an individual genomic (pre-called) variant calling format (GVCF) file for an individual sample. Next, we will use the GVCF file to identify the lineage of your sample using *‘mtbtyper’*.

# Creating index and dictionary files for the reference genome

Similar to what we did in the previous practical, we first need to create the reference genome index, required for GATK to efficiently process the data.

Go to the directory

***‘/home/[user]/mtb\_wgs\_analysis\_workshop/ref/’***

by using ‘*cd*’, and then run the following commands:

**$ samtools faidx reference.fasta**

**$ gatk CreateSequenceDictionary -R reference.fasta**

These will create the index and dictionary files, denoted by the ‘*.fai*’ and ‘*.dict*’ suffixes, respectively.

# Per-sample variant calling

Similar to other bioinformatics tools that we have already learned, we can type the name of any GATK tool into the command line and press ‘*Enter*’ to see its available options. For example, with the command ‘*gatk HaplotypeCaller*’, it will return the tool description and version as well as a long list of available arguments and their descriptions. The top of the output looks like the figure below:

A screenshot of a computer program

Description automatically generated

Next, we will do ‘*per-sample variant calling*’ for our Mtb sequence data from the BAM read alignment files by using ‘*GATK HaplotyperCaller*’. Apart from specifying the paths and names of the reference genome, input, and output files, we have to tell the program to analyse our data as a haploid genome, and record non-variant sites as condensed blocks in the output files in a GVCF format, and call variant only when those bases have quality score >20. To do this, first change your current directory to

*‘****/home/[user]/mtb\_wgs\_analysis\_workshop/’***

and run the following command.

*gatk HaplotypeCaller \*

*-I /bam/sample\_01.sorted.bam \*

*-O /per-sample\_gvcf/sample\_01.g.vcf.gz \*

*-R /reference/reference.fasta \*

*--sample-ploidy 1 \*

*--emit-ref-confidence GVCF \*

*--min-base-quality-score 20*

This command will generate ‘*sample\_01.g.vcf.gz’* file, which is a zipped GVCF file of the sample *‘sample\_01’* from the read mapping alignment file *‘sample\_01.sorted.bam’*.

# MTB lineage identification with mtbtyper

*‘mtbtyper’* is among the best tools that are developed for typing MTBC lineage. It is compact, fast, and provides one of the most comprehensive databases of MTBC lineage- and sub-lineage specific markers. mtbtyper is publicly available at [mtbtyper](https://github.com/ythaworn/mtbtyper). Currently, the program accepts input files in GVCF format (and also VCF, see the next practical), which can be generated by using variant calling pipelines such *‘snpplet’*, *‘TB-profiler’*, as well as the work flow that we used in previous steps.

*‘mtbtyper’* has not been installed on your computer yet. We will install it together by following the instructions below.

*pip install numpy pandas scikit-allel*

*cd /mtb\_wgs\_analysis\_workshop/*

***git clone*** [***https://github.com/ythaworn/mtbtyper.git***](https://github.com/ythaworn/mtbtyper.git)

Now, you should have *‘mtbtyper’* in the directory ‘*mtb\_wgs\_analysis\_workshop*’.

Next, we will do lineage identification. First, go to the directory

***‘mtb\_wgs\_analysis\_workshop/result/per-sample\_gvcf/’***

And then type the following commands:

*python /home/[user]/mtb\_wgs\_analysis\_workshop/mtbtyper/mtbtyper.py \*

*–snpdb /home/[user]/mtb\_wgs\_analysis\_workshop/mtbtyper/snpdb \*

*-f lineage.csv \*

*./*

We will now have an output file named *‘lineage.csv’* in the current working directory. This file can be opened with any text editor or MS Excel. We will explore the result together during the session.