Module 7

Variant Calling

Goal: Learn how to use various tools to identify variants after re-sequencing

Input(s): `Mo_FR13_IP{1-3}_accepted_hits.bam`

Output(s): `Mo_FR13_alignments_merged.bam`
`Moryzae_FR13_alignments.bcf`
`Moryzae_FR13_alignments.vcf`

Three of the RNA samples used for RNAseq analysis were generated from fungal strain FR13, which is expected to show several sequence differences relative to the 70-15 strain that was used to generate the reference genome. Therefore, we will use the RNAseq alignments to search for nucleotide differences between FR13 and the 70-15 reference strain.

7.1 Merge the FR13 alignment (.bam) files

Having confidence in variant calls requires one to have multiple reads aligned across the region where there is a genetic difference between the sample and the reference. When sequencing DNA, this is easily accomplished by ensuring that we have enough sequence data to provide several-fold coverage of the genome (≥ 20x). However, with RNAseq data, we have no control over coverage because this is determined by the expression level of each gene at the time and place where the RNA was extracted. Therefore, to maximize coverage across each gene for the purpose of our analysis, we will merge the alignments file for the three *in planta* RNA replicates.

☐ We will use the **samtools** merge tool. (Note that bam files must be sorted by coordinates before they are merged. We already sorted the files in the *rnaseq* lab, so we can use them directly here.)

- `samtools merge Mo_FR13_alignments_merged.bam ../rnaseq/alignments/*FR13*bam`

- ☐ Take a quick look at the *.bam* file. Is it in binary format? You can take a look at it by using **samtools** again:

```
• samtools view Mo_FR13_alignments_merged.bam
```

- ☐ Whoa, can you read the output? Quit the process (control-c) and re-run the command with a pipe into another program that will enable you to look at a few lines at a time.
- ☐ Does the output still look like it contains sam alignment data? If so, proceed to the following.
- ☐ We must provide **mpileup** with the reference genome that was used for alignment. To speed up its computations, **mpileup** uses an indexed version of the genome. This index is different to the one generated with **bowtie2**, and we will use **samtools** to generate it.

```
• samtools faidx
  ../blast/magnaporthe_oryzae_70-15_8_supercontigs.fasta
```

  This will create an index file named *magnaporthe_oryzae_70-15_8_supercontigs.fasta.fai.*

- ☐ Check to make sure the index was created. (Note that the file is created in the same directory as the reference genome.)

7.2 Perform the variant calling

We will use the **bcftools mpileup** utility to extract information on nucleotide variations (substitutions, insertions, deletions) between our sequence sample and a reference genome. **mpileup** does this by using information contained in the CIGAR (Concise Idiosyncratic Gapped Alignment Report) string and the BTOP (back track operations) string in the MD:Z: field of the *.sam* (or *.bam*)file.

Usage:

```
bcftools mpileup [options] -f <reference_genome> <BAM alignment(s)>
```

- ☐ Make sure you are using **screen**.
- ☐ Run **mpileup** on the sorted file.

```
• bcftools mpileup --threads 2 -f
  ../blast/magnaporthe_oryzae_70-15_8_supercontigs.fasta
  Mo_FR13_alignments_merged.bam
```

`--threads` number of processors to use

`-f [FILE]` name of reference genome

bcftools, like **samtools**, outputs results to the screen, making the output kind of hard to read. However, at least you can see that the program is doing something interesting.

- ☐ Stop the process (using control-c) so you don't have to wait a LONG time for the output to finish printing to the screen.

- ☐ Re-run the previous command but redirect the results to a file called *Mo_FR13_alignments.vcf*. (**Note**: the program will take several minutes to complete, during which time you will receive no progress updates—all program outputs are directed to the specified output file.)
- ☐ Inspect the *Mo_FR13_alignments.vcf* file. This summarizes variant statistics for every position in the reference genome for which there are aligned reads. However, it does contain variant calls.
- ☐ We will use **bcftools** to call the variants. **bcftools** is like **samtools** in that it sends results to the screen so we need to re-direct the output to a file.

```
• bcftools call -v -c --ploidy 1 Mo_FR13_alignments.vcf >
  Mo_FR13_alignments_called.vcf
```

`-v`	output potential variant sites only (i.e., skip monomorphic ones)
`-c`	call variants using the original method implemented in **samtools mpileup**
`--ploidy`	set to 1 because *Magnaporthe* is haploid, and we only expect one copy of each gene unless the sequence is repeated

7.3 Examine the resulting variant calls

- ☐ Inspect the *Mo_FR13_alignments_called.vcf* file.

At the head of the file is some information on how to interpret the various fields. Below, each line provides information on a predicted variant at a specific nucleotide position within the chromosome. Here you should be able to recognize data that make sense.

Unfortunately, the header does not provide much information on the overall structure of the ***.vcf*** file. The main fields in the tab-delimited section are as follows:

Column 1: Chromosome number

Column 2: Nucleotide position

Column 3: SNP ID (if previously characterized and named)

Column 4: Nucleotide in reference genome

Column 5: Alternate allele(s) identified in sequence reads

Column 6: Quality of SNP call

Column 7: Filtering information (“.” = no filter; “Low Qual”; or “PASS”)

Column 8: SNP information (see list of INFO fields in file header)

Columns 9 & 10: SNP formats (see the list of FORMAT fields in file header)

A complete description of the VCF format is in the VCFv4.1.pdf file on the Canvas site.

Column 10 in the *Mo_FR13_alignments_called.vcf* file contains information that informs us about the likelihood that a given SNP (or INDEL) is valid. First, to have confidence in the call, we want to have several reads that support it, and because we are working with a haploid fungus, ideally we want all reads that overlap a given site to have the alternate (variant) allele. Unfortunately, variant callers are not very smart when operating in haploid mode (they were mostly designed for human variant calling). As a result, they often call variants at sites that are clearly heterozygous for the reference/alternate alleles. Heterozygosity in a haploid organism indicates that there are two or more copies of the site being interrogated but, more importantly, it tells us that the SNP is likely to be false. There are many instances of heterozygosity for variant calls in the *Mo_70-15_FR13.vcf* file, so we need to filter the file to remove these suspect calls.

- ☐ First, use **more** to look at a few pages of the vcf file and pay attention to the DP4= field in column 10. (Note: this is a w-i-d-e column!) There are four values that follow DP4=: forward strand reads supporting ref allele, reverse strand reads supporting ref allele, forward strand reads supporting the alt allele, reverse strand reads supporting the alt allele. For a true SNP, there should be no reads at all supporting the ref allele.
- ☐ Use grep to count the total number of SNP calls (don't count INDELs). Next, count the number of low quality calls where there are some reads supporting the reference allele, position, and DP4= values for one such suspect record.
  Total SNP calls: _____; Low quality (suspect) SNPs: _____

In addition, many of the SNPs that were called were supported by very few reads, and so we can't be confident that these variant calls aren't due to sequencing errors. Therefore, we will filter the SNP calls to retain only those calls that are supported by at least 10 reads, with no reads containing the ref allele.

- ☐ There are specific tools for this purpose, but why waste our time trying to remember the necessary commands and options when we can easily do this with **grep**?

```
• grep DP=[1-9][0-9] Mo_FR13_alignments_called.vcf | grep DP4=0,0,
  > Mo_FR13_alignments_filtered.vcf
```

`grep DP=[1-9][0-9]` look for "`DP=`" followed by any number between 1 and 9, followed by any number between 0 and 9. (This is how we make sure that `DP=` is followed by a number that is equal to, or greater than, 10.)

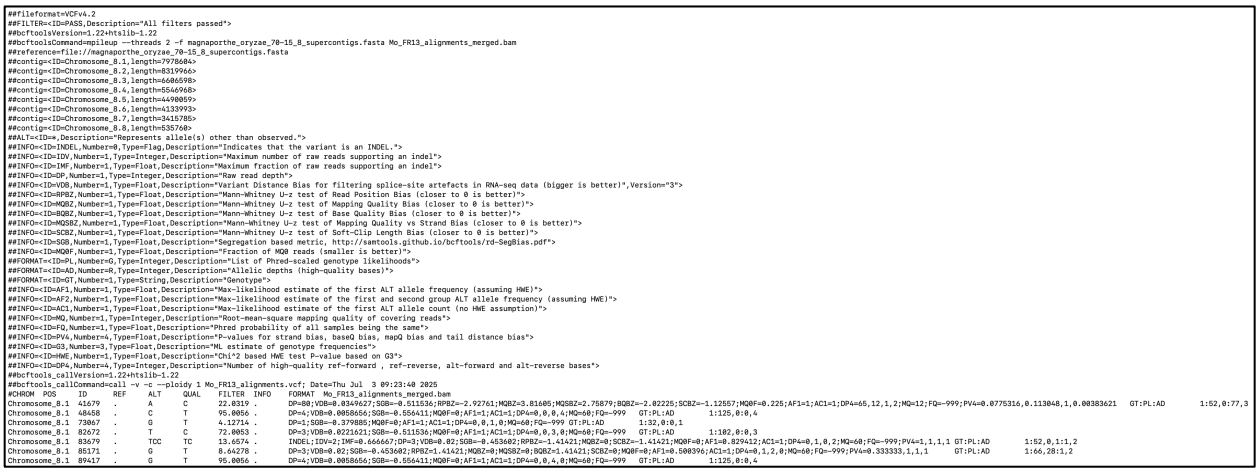
Examine the resulting filtered file containing "high quality" variant calls, and use the suggested tools with pipes to answer the following questions:

- ☐ How many SNPs were identified on Chromosome 8.1? (**grep**)_____
- ☐ How many INDELs were found on Chromosome 8.1? (**grep**) _____
- ☐ Are there any indels that are greater than one nucleotide in length? (**awk {print length(...)}, sort, uniq)**" yes__; no __
- ☐ What is the greatest depth of coverage (DP=) for a SNP in the filtered dataset? (**awk**, **sed**, **sort**, **uniq**) _____

Appendix

The variant call format

This is a standard text file format used in **bioinformatics** for storing information about **genetic variants** (like SNPs, insertions, deletions, etc.) identified in DNA sequencing data. It has a header section (lines starting with ##) which documents the mpileup and variant call commands that were used. This also provides information about the reference genome used, as well as the various data "tags" that describe the data. This is followed by the data section which lists the sites on the reference where variants have been found as well as various statistics that support each call. These statistics are denoted with shorthand tags, short descriptions of which can be found in the header.



```
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##bcftoolsVersion=1.22+htslib-1.22
##bcftoolsCommand=mpileup --threads 2 -f magnaporthe_oryzae_70-15_8_supercontigs.fasta Mo_FR13_alignments_merged.bam
##reference=file://magnaporthe_oryzae_70-15_8_supercontigs.fasta
##contig=<ID=Chromosome_8.1,length=7978604>
##contig=<ID=Chromosome_8.2,length=8319966>
##contig=<ID=Chromosome_8.3,length=6606598>
##contig=<ID=Chromosome_8.4,length=5546968>
##contig=<ID=Chromosome_8.5,length=4490059>
##contig=<ID=Chromosome_8.6,length=4133993>
##contig=<ID=Chromosome_8.7,length=3415785>
##contig=<ID=Chromosome_8.8,length=535760>
##ALT=<ID=*,Description="Represents allele(s) other than observed.">
##INFO=<ID=INDEL,Number=0,Type=Flag,Description="Indicates that the variant is an INDEL.">
##INFO=<ID=IDV,Number=1,Type=Integer,Description="Maximum number of raw reads supporting an indel">
##INFO=<ID=IMF,Number=1,Type=Float,Description="Maximum fraction of raw reads supporting an indel">
##INFO=<ID=DP,Number=1,Type=Integer,Description="Raw read depth">
##INFO=<ID=VDB,Number=1,Type=Float,Description="Variant Distance Bias for filtering splice-site artefacts in RNA-seq data (bigger is better)",Version="3">
##INFO=<ID=RPBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Read Position Bias (closer to 0 is better)">
##INFO=<ID=MQBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Mapping Quality Bias (closer to 0 is better)">
##INFO=<ID=BQBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Base Quality Bias (closer to 0 is better)">
##INFO=<ID=MQSBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Mapping Quality vs Strand Bias (closer to 0 is better)">
##INFO=<ID=SCBZ,Number=1,Type=Float,Description="Mann-Whitney U-z test of Soft-Clip Length Bias (closer to 0 is better)">
##INFO=<ID=SGB,Number=1,Type=Float,Description="Segregation based metric, http://samtools.github.io/bcftools/rd-SegBias.pdf">
##INFO=<ID=MQ0F,Number=1,Type=Float,Description="Fraction of MQ0 reads (smaller is better)">
##FORMAT=<ID=PL,Number=G,Type=Integer,Description="List of Phred-scaled genotype likelihoods">
##FORMAT=<ID=AD,Number=R,Type=Integer,Description="Allelic depths (high-quality bases)">
##FORMAT=<ID=GT,Number=1,Type=String,Description="Genotype">
##INFO=<ID=AF1,Number=1,Type=Float,Description="Max-likelihood estimate of the first ALT allele frequency (assuming HWE)">
##INFO=<ID=AF2,Number=1,Type=Float,Description="Max-likelihood estimate of the first and second group ALT allele frequency (assuming HWE)">
##INFO=<ID=AC1,Number=1,Type=Float,Description="Max-likelihood estimate of the first ALT allele count (no HWE assumption)">
##INFO=<ID=MQ,Number=1,Type=Integer,Description="Root-mean-square mapping quality of covering reads">
##INFO=<ID=FQ,Number=1,Type=Float,Description="Phred probability of all samples being the same">
##INFO=<ID=PV4,Number=4,Type=Float,Description="P-values for strand bias, baseQ bias, mapQ bias and tail distance bias">
##INFO=<ID=G3,Number=3,Type=Float,Description="ML estimate of genotype frequencies">
##INFO=<ID=HWE,Number=1,Type=Float,Description="Chi^2 based HWE test P-value based on G3">
##INFO=<ID=DP4,Number=4,Type=Integer,Description="Number of high-quality ref-forward , ref-reverse, alt-forward and alt-reverse bases">
##bcftools_callVersion=1.22+htslib-1.22
##bcftools_callCommand=call -v -c --ploidy 1 Mo_FR13_alignments.vcf; Date=Thu Jul  3 09:23:40 2025
#CHROM  POS     ID      REF     ALT     QUAL    FILTER  INFO    FORMAT  Mo_FR13_alignments_merged.bam
Chromosome_8.1  41679   .       A       C       22.0319 .       DP=80;VDB=0.0349627;SGB=-0.511536;RPBZ=-2.92761;MQBZ=3.81605;MQSBZ=2.75879;BQBZ=-2.02225;SCBZ=-1.12557;MQ0F=0.225;AF1=1;AC1=1;DP4=65,12,1,2;MQ=12;FQ=-999;PV4=0.0775316,0.113048,1,0.00383621   GT:PL:AD        1:52,0:77,3
Chromosome_8.1  48458   .       C       T       95.0056 .       DP=4;VDB=0.0058656;SGB=-0.556411;MQ0F=0;AF1=1;AC1=1;DP4=0,0,0,4;MQ=60;FQ=-999   GT:PL:AD        1:125,0:0,4
Chromosome_8.1  73867   .       G       T       4.12714 .       DP=1;SGB=-0.379885;MQ0F=0;AF1=1;AC1=1;DP4=0,0,1,0;MQ=60;FQ=-999 GT:PL:AD       1:32,0:0,1
Chromosome_8.1  82672   .       T       C       72.0053 .       DP=3;VDB=0.0221421;SGB=-0.511536;MQ0F=0;AF1=1;AC1=1;DP4=0,0,3,0;MQ=60;FQ=-999   GT:PL:AD        1:102,0:0,3
Chromosome_8.1  83679   .       TCC     TC      13.6574 .       INDEL;IDV=2;IMF=0.666667;DP=3;VDB=0.02;SGB=-0.453602;RPBZ=-1.41421;MQBZ=0;SCBZ=-1.41421;MQ0F=0;AF1=0.829412;AC1=1;DP4=0,1,0,2;MQ=60;FQ=-999;PV4=1,1,1,1 GT:PL:AD        1:52,0,1:1,2
Chromosome_8.1  85171   .       G       T       8.64278 .       DP=3;VDB=0.02;SGB=-0.453602;RPBZ=1.41421;MQBZ=0;MQSBZ=0;BQBZ=1.41421;SCBZ=0;MQ0F=0;AF1=0.500396;AC1=1;DP4=0,1,2,0;MQ=60;FQ=-999;PV4=0.333333,1,1,1      GT:PL:AD        1:66,28:1,2
Chromosome_8.1  89417   .       G       T       95.0056 .       DP=4;VDB=0.0058656;SGB=-0.556411;MQ0F=0;AF1=1;AC1=1;DP4=0,0,4,0;MQ=60;FQ=-999   GT:PL:AD        1:125,0:0,4
```

The various data columns are as follows:

CHROM	Chromosome (e.g., chr1)
POS	Position on the chromosome
ID	Variant ID (e.g., official rsID in dbSNP)
REF	Reference base(s)
ALT	Alternate base(s)
QUAL	Quality score for the variant (PHRED scale)
FILTER	Filter status (e.g., PASS or reason for filtering out)
INFO	Additional information (e.g., read depth: DP=100; BQBZ=base quality bias)
FORMAT	Format of the genotype fields (e.g., GT:PL:AD = genotype:phred likelihood:allelic depths)
sample1, sample2...	Genotypes for each sample (0=ref; 1=alt; 0/1 =heterozygous). For our analysis we only have one genotype column but when there are multiple samples, there will be one genotype fields per sample.