Applications of Bioinformatics in Plant Breeding
Practical: Variant Discovery in NGS data

In this practical we will use our Linux environment to discover and filter variants from a previously produced BAM file, and then visualise and examine these in Tablet.

This requires the following data files, all of which should be in your shared Windows/Linux data directory:

- reference_H35_varCallingPractical.fasta
- reference_H35_varCallingPractical.fasta.fai
- variantCallingPractical_barley_iSelect.bam
- variantCallingPractical_barley_iSelect.bam.bai

If you wish, you can copy and paste the commands as printed here (grey text boxes) straight into the command line.

----

1. Run Freebayes on defaults to discover variants

navigate to the data directory and then execute freebayes from there:

```
freebayes -f reference_H35_varCallingPractical.fasta -b
variantCallingPractical_barley_iSelect.bam -v SNPs_defaults.vcf --ploidy 2
```

This should produce 33 variants.

- Open the resulting VCF

----

2. Freebayes with slightly more stringent filtering

First, run Freebayes with the –help option to see a list of all the configurable options:

```
freebayes -help
```

Then run Freebayes with this added option for filtering:

```
freebayes -f reference_H35_varCallingPractical.fasta -b
variantCallingPractical_barley_iSelect.bam -v SNPs_prefiltered.vcf --ploidy
2 --min-alternate-count 3
```

The latter option “--min-alternate-count 3” applies a filter that requires *at least* one sample at a candidate variant location to have at least 3 reads with the alternate allele.

- Open the resulting VCF file and inspect it.

- How many variants do we have now?

----

3. Post-Freebayes filtering

We will use the following tools to do this in two stages:

- `cat` – a built-in Linux tool that will read a whole file to the standard output stream in one go
- `vcf-annotate` – a tool from the vcftools package (vcftools.sourceforge.net) which annotates the "FILTER" field in a VCF file with information whether a variant is "PASS" or whether it has failed filtering on one or more counts (these all get listed in this field)
- `awk` -- a built-in Linux tool for text processing; here, we use this to filter for those lines in the VCF file coming from vcf-annotate that have a value of "PASS" in column 7 ($7=="PASS")

```
cat SNPs_prefiltered.vcf | vcf-annotate -f Q=20/d=100 > SNPs_annotated.vcf

cat SNPs_annotated.vcf | awk '$7=="PASS"' > SNPs_filtered.vcf
```

This should further reduce the number of variants from the previous step. SNPs with a SNP quality of less than 20 (`Q=20`) and/or a total read coverage of less than 100 (`d=100`) have been filtered out, and as a result our set of SNPs is more robust.

Inspect the annotated VCF file in Excel (SNPs_annotated.vcf). Look at the "FILTER" column. This should tell you for each variant whether it passed or what it failed on. In this file, the failing SNPs have not been removed yet, just annotated. It is handy to keep this for inspection though.

N.B. The final VCF file (SNPs_filtered.vcf) here does not contain a header, due to using awk for filtering.

----

4. Convert the VCF format files to GFF format (a feature annotation format)

This will allow us to import the variants into Tablet as features so we can view them as annotation of our mappings.

We can do this both for the filtered and unfiltered VCF file. If we then import these into Tablet we can generate two annotation tracks so that we can see what effect the filtering has had.

The conversion is done with Java code provided for the purpose of this practical. The syntax for the command is:

```
java ConvertVCFToGFF <vcf file> <trackAnnotation>
```

First, convert the unfiltered variants from our default run:

```
java ConvertVCFToGFF SNPs_defaults.vcf unfiltered
```

Then , do the same for the filtered variants:

```
java ConvertVCFToGFF SNPs_filtered.vcf filtered
```

This should results in two new files called SNPs_defaults.gff and SNPs_filtered.gff.

Note that in the case of indels the positions stated by Freebayes may differ from those stated by Tablet. This is because Freebayes usually reports indels as complex haplotypes with some bases either side of the actual indel itself.

- Find locations that are present in the unfiltered set of variants but are missing from the filtered set.
- Refer to their entry in the original VCF file.
- Why have they been removed? (Tip: the read depth/coverage value in VCF file is stored in the INFO field, against a tag called "DP").