Samtools, con't (mpileup)

Pileups with samtools pileup, variant calling, and base alignment quality

- **We continue working with samtools, and in this case, will be working with the pileup format, which is a plain-text format that summarizes reads' bases at each chromosome position by stacking or "piling up" aligned reads.**
- **The per-base summary of the alignment data created in a pileup can then be used to identify variants (regions different from the reference) and determine genotypes**

Today's lecture

1. Use mpileup subcommand to create pileups from BAM files, the first step in samtools-based variant calling pipelines
2. Examine misalignments can lead to erroneous base calls
3. Examine the base alignment quality algorithm which can prevent erroneous variant calls due to misalignment

Prepare workspace

Log-on to flux using an interactive shell

```
$   qsub -I -V -l procs=2 -l walltime=3:00:00 -A eeb416f15_flux -l qos=flux -q
flux
```

Load the samtools module and the bcftools module

```
$   module add med samtools/1.2 bcftools/1.2
```

Input

1. samtools pileup requires an input BAM file and a reference in fast format
2. We're using the --region call to limit our pileup to the same region we looked at last class period
3. We disable the BAQ command, or Base Alignment Quality

These pileups are the per-position summary of the data in aligned reads

```
$   samtools mpileup --no-BAQ --region 1:215906528-215906567 --fasta-ref
human_g1k_v37.fasta NA12891_CEU_sample.bam
```

Using version 1.19:

```
$   samtools mpileup -B -r 1:215906528-215906567 -f human_g1k_v37.fasta
NA12891_CEU_sample.bam
```

Output

1. Col 1--Reference sequence name
2. Col 2--Position in reference sequence
3. Col 3--Reference sequence base at this position
4. Col 4--depth of aligned reads at this position
5. Col 5--Encodes the reference reads bases.
    - (.) indicate a reference sequence match on the forward strand, (,) indicate ref sequence match on the reverse strand.
    - Uppercase AGTCN indicates a mismatch on the forward strand, agtcn indicates mismatch on reverse strand
    - The ^ and $ characters indicate the start and end of reads
    - Insertions are denoted with a + sign, deletions are -
6. Col 6--Mapping quality of each alignment is specified after the beginning of the

alignment character, ^ as the ASCII character value minus 33.

While the above pileup call gives us the per-position summary of the data in aligned reads, if we want to infer variant and genotype calls (we generally DO!), we have to make inferences about noisy alignment data.

Most variant calling use probabilistic frameworks to make reliable inferences of genotype calls given low coverage, poor base quality, possible misalignments, and other issues

Calling variants with samtools and bcftools is a two-step process.

1. **First, samtools mpileup called with -v or -g arguments will generate genotype likelihoods for every site in the genome. These calls will be returned in a vcf flat file (variant call format).**
2. **Second, bcftools will filter these results so only variant sites remain**

Command

```
$   samtools mpileup --no-BAQ --region 1:215906528-215906567 --fasta-ref
human_g1k_v37.fasta NA12891_CEU_sample.bam > NA12891_CEU_sample.vcf.gz
```

Using version 1.19:

```
$   samtools mpileup -B -r 1:215906528-215906567 -f human_g1k_v37.fasta
NA12891_CEU_sample.bam > NA12891_CEU_sample.vcf
```

This produces a zipped vcd file, which contains intermediate variant and genotype

data.

```
$   zgrep "^##" -v NA12891_CEU_sample.vcf | awk 'BEGIN{OFS="\t"} {split($8, a,
";"); print $1,$2,$4,$5,$6,a[1],$9,$10}'
```

We then take these intermediate variant calls and pass them to bcftools call, which uses information from mpileup to make an inference whether sites are really variant and which each ind's genotype is.

```
$   bcftools call -v -m NA12891_CEU_sample.vcf.gz >
NA12891_CEU_sample_calls.vcf.gz
```

Above we used -m which specifies multi allelic caller (could have used -c, for consensus). The -v option outputs only variant sites.

```
$   zgrep "^##" -v NA12891_CEU_sample_calls.vcf.gz | awk 'BEGIN{OFS="\t"}
{split($8, a, ";"); print $1,$2,$4,$5,$6,a[1],$9,$10}'
```

bcftools call has an estimated quality score for each alternative alleles in ALT (Phred-scaled values that estimate the probability that the alternative allele is incorrect). Higher qual score indicates higher likelihood site is a variant.

Below we are running the same call as above, but omitting the -v so that we can view all of the potential variants

```
$   bcftools call -m NA12891_CEU_sample.vcf.gz | grep -v "^##" | awk
'BEGIN{OFS="\t"} {split($8, a, ";"); print $1,$2,$4,$5,$6,a[1],$9,$10}'
```

Now we want to determine if our 'variants' are marked as such because

of a bad alignment rather than true genotypic variation

```
$   samtools mpileup -u -v --region 1:215906528-215906567 --fasta-ref
human_g1k_v37.fasta NA12891_CEU_sample.bam > NA12891_CEU_sample-baq.vcf.gz
```

Above we enable the Base Alignment Quality option

```
$   grep "^##" -v NA12891_CEU_sample_calls.vcf.gz | awk 'BEGIN{OFS="\t"}
{split($8, a, ";"); print $1,$2,$4,$5,$6,a[1],$9,$10}'
```

Note that we have lost the potential variants at ...547 and ...548. The BAQ algorithm down weighted the bases around this low complexity regions such that samtools mpileup no longer considers them true invariant sites.