Back to BedTools – most important tool: intersect

- Some tools are easier to rewrite: slop, flank, complement (although can be tricky)

- Intersect is the bread-and-butter, It would take substantial work to write a better intersect than what bedtools has
Using the genes feature file

```
allertsporthos -/work/lec20
# head -J simple.gff
chr1 SGD chromosome 1 230218 . . .
chr1 SGD telomeric_repeat 1 62 . . .
chr1 SGD telomere 1 801 . . .

allertsporthos -/work/lec20
# cat simple.gff | awk '{print $0}' > genes.gff
```

Running the intersection

```
allertsporthos -/work/lec20
$ ~/bin/bedtools intersect -i bwa.bam -b genes.gff -s -v > genes.bed
```

-wo flag writes entry A then entry B followed by the overlap

Quite a few features – strand, overlap fraction and others

```
```

Intersect variants: window, closest

- **bedtools window**

Examine a "window" around each feature in A and reports all features in B that overlap the window.

- **bedtools closest**

For each feature in A, finds the closest feature (upstream or downstream) in B.

**Window**: extend or flank the original intervals and use intersect.

**Closest**: intersect then sort by gene name then by overlap.
Other helpful bedtool commands:

**getfasta** — extracts sequences

```
getfasta [OPTIONS] -fi <fasta> [-bed <bed>]
```

**ge:** extracts sequences

```
bed tools getfasta (aka fastaFromBed)
```

**getseq**

```
getseq [OPTIONS] -fi <fasta> [-bed <bed>]
```

**Group on a column and applies an operator to a different column**

**bedtools groupby**

```
bedtools groupby
```

**get the sequence to every feature annotated as binding_site**

```
get the sequence to every feature annotated as binding_site
```

```
get the sequence to every feature annotated as binding_site
```
More on groupby

- data must be sorted on the grouping column
- Valid operators: sum, mean, median, mode, min, max (see help for more)
- Example: Find the average length of each feature in the `features.gff` file

Homework 20: refine homework 17

- Did you get fewer or more SNPs than expected?
- Current homework will be to match the called SNPs to the expectations

The full data script (see website).
1. Generates mutated data
2. Aligns mutated data
3. Creates the BAM file
4. Generates genotype information
5. Calls the SNPs

Question

Which SNPs match the expectation?
Which SNPs do not?
Transform the expected SNPs output

- wgsim mutation output does not produce a known interval format → transform it to one.
- What should it be? Ideally a VCF but that may be too complicated for our needs
- Keep it simple → we could also turn it into something simple like GFF

Generate a GFF file from the mutations

Intersect the two files

Homework 20

Refine homework 17. Use the VCF file and mutations file from that homework. Show the code and report the numbers for

- TP (True Positives): number of expected SNPs that you have found
- FP (False Positives) – SNPs that you found but are not in the mutation file
- FN (False Negatives) – SNPs that you should have found but did not

Look at the FP and FN list, in one paragraph discuss of some common aspects of the data that ended up on the FP and FN list.