Project 2

For this project, it is recommended that you use the virtual environment provided with the Course package and the tools therein. You may also use your own system and software, however make sure that appropriate versions are installed. The answers are compatible with the following versions of the software: samtools v.1.2, bedtools v.2.24.0.

As part of a larger project cataloging genetic variation in the plant Arabidopsis thaliana, you sequenced and assembled the genome of one strain ('wu_0_A'), then mapped back the reads to the assembled genome. The resulting BAM file is included in the package 'gencommand_proj2_data.tar.gz'. Using SAMtools and BEDtools as well as other Unix commands introduced in this course, examine the files and answer the following questions. NOTE: Input data have been obtained and modified from those generated by the 1001 Genomes Project, accession 'Wu_0_A'.

Apply these rules and steps to the questions marked above each rule.

Questions 1 - 5:
For the original set of alignments (file 'athal_wu_0_A.bam'):

1. How many alignments does the set contain?

```
(base) [root@522cc4c0d20e proj2]# samtools flagstat athal_wu_0_A.bam
221372 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
0 + 0 supplementary
20419 + 0 duplicates
218469 + 0 mapped (98.69%:-nan%)
155851 + 0 paired in sequencing
77895 + 0 read1
77956 + 0 read2
142663 + 0 properly paired (91.54%:-nan%)
150045 + 0 with itself and mate mapped
2903 + 0 singletons (1.86%:-nan%)
4938 + 0 with mate mapped to a different chr
2120 + 0 with mate mapped to a different chr (mapQ>=5)
(base) [root@522cc4c0d20e proj2]#
```

   a.
   b. 221372
2. How many alignments show the read's mate unmapped?
   a. Unmapped reads: column 7, *
   b.

```
(base) [root@522cc4c0d20e proj2]# samtools view athal_wu_0_A.bam | cut -f7 | grep -c '*'
65521
```

   c. **65521**
3. How many alignments contain a deletion (D)?
   a. Deletions: column 6 (cigar), D

```
(base) [root@522cc4c0d20e proj2]# samtools view athal_wu_0_A.bam | cut -f6 | grep -c 'D'
2451
```

b. 2451
c. **2451**

4. How many alignments show the read's mate mapped to the same chromosome?
    a. Mates mapped: column7, =

```
(base) [root@522cc4c0d20e proj2]# samtools view athal_wu_0_A.bam | cut -f7 | grep -c '='
150913
```

    b. 150913
    c. **150913**
5. How many alignments are spliced?
    a. Spliced: column6 (cigar), N

```
(base) [root@522cc4c0d20e proj2]# samtools view athal_wu_0_A.bam | cut -f6 | grep -c 'N'
0
```

    b. 0
    c. **0**

Questions 6 - 10:
Extract only the alignments in the range "Chr3:11,777,000-11,794,000", corresponding to a locus of interest. For this alignment set:

Before analysis we have to sort and index the bam file

```
(base) [root@522cc4c0d20e proj2]# samtools sort athal_wu_0_A.bam data.sorted
(base) [root@522cc4c0d20e proj2]# samtools index data.sorted.bam
```

6. How many alignments does the set contain?

```
(base) [root@522cc4c0d20e proj2]# samtools view -b data.sorted.bam 'Chr3:11777000-11794000' > data_region.bam
(base) [root@522cc4c0d20e proj2]# samtools flagstat data_region.bam
7081 + 0 in total (QC-passed reads + QC-failed reads)
0 + 0 secondary
0 + 0 supplementary
606 + 0 duplicates
6598 + 0 mapped (93.18%:-nan%)
5098 + 0 paired in sequencing
2545 + 0 read1
2553 + 0 read2
3568 + 0 properly paired (69.99%:-nan%)
4132 + 0 with itself and mate mapped
483 + 0 singletons (9.47%:-nan%)
428 + 0 with mate mapped to a different chr
366 + 0 with mate mapped to a different chr (mapQ>=5)
```

    a.
    b. **7081**
7. How many alignments show the read's mate unmapped?

```
(base) [root@522cc4c0d20e proj2]# samtools view data_region.bam | cut -f7 | grep -c '*'
1983
```

    a.
    b. **1983**
8. How many alignments contain a deletion (D)?

```
(base) [root@522cc4c0d20e proj2]# samtools view data_region.bam | cut -f6 | grep -c 'D'
31
```

    a.
    b. **31**
9. How many alignments show the read's mate mapped to the same chromosome?

```
(base) [root@522cc4c0d20e proj2]# samtools view data_region.bam | cut -f7 | grep -c '='
4670
```

    a.
    b. **4670**
10. How many alignments are spliced?

```
(base) [root@522cc4c0d20e proj2]# samtools view data_region.bam | cut -f6 | grep -c 'N'
0
```

    a.

b. **0**

Questions 11 - 15:
Determine general information about the alignment process from the original BAM file.

11. How many sequences are in the genome file?

```
(base) [root@522cc4c0d20e proj2]# samtools view -H athal_wu_0_A.bam | grep -c 'SN:'
```

a. 7
b. **7**

12. What is the length of the first sequence in the genome file?

```
(base) [root@522cc4c0d20e proj2]# samtools view -H athal_wu_0_A.bam | grep  'SN:' | more
@SQ     SN:Chr1 LN:29923332     AS:wu_0.v7.fas  SP:wu_0.v7.fas
@SQ     SN:Chr2 LN:19386101     AS:wu_0.v7.fas  SP:wu_0.v7.fas
@SQ     SN:Chr3 LN:23042017     AS:wu_0.v7.fas  SP:wu_0.v7.fas
@SQ     SN:Chr4 LN:18307997     AS:wu_0.v7.fas  SP:wu_0.v7.fas
@SQ     SN:Chr5 LN:26567293     AS:wu_0.v7.fas  SP:wu_0.v7.fas
@SQ     SN:chloroplast  LN:154478       AS:wu_0.v7.fas  SP:wu_0.v7.fas
@SQ     SN:mitochondria LN:366924       AS:wu_0.v7.fas  SP:wu_0.v7.fas
```

a.
b. 29923332

13. What alignment tool was used?

```
(base) [root@522cc4c0d20e proj2]# samtools view -H athal_wu_0_A.bam | grep  '^@PG'
@PG     ID:stampy       VN:1.0.3_(r627) CL:-g /lustre/scratch103/sanger/xcg/tmp/tmp.zYfXz26246 -h /lustre/scratch103/sanger/xcg/tmp/tmp.zYfXz26246 --readgro
up=ID:Wii_PER01,LB:wu_phaseI,SM:wu_0,PI:400,PL:SLX,DS:wu_0_Genome --bwaoptions=-q10 /lustre/scratch103/sanger/xcg/wu_0.v7.fas -o /lustre/scratch103/sanger/x
cg/wu_0/A/aln_A1.sp0.sam -M /lustre/scratch103/sanger/xcg/wu_0/read_1_1.sp0.fq.gz /lustre/scratch103/sanger/xcg/wu_0/read_1_2.sp0.fq.gz
```

a.
b. Stampy

14. What is the read identifier (name) for the first alignment?

```
(base) [root@522cc4c0d20e proj2]# samtools view athal_wu_0_A.bam | head -1
GAII05_0002:1:113:7822:3886#0   1187    Chr3    11699950        60      51M     =       11700332        433     AAAAAAAATGTAAAACTGCTAAATCTCTCCTCTCTAAAGAACTC
GTCCCCG CCCCCCBBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBAAB??@ACBBCCCD     PQ:i:21 SM:i:37 UQ:i:0  MQ:i:37 XQ:i:0  RG:Z:H100223_GAII05_0002
```

a.
b. GAII05_0002:1:113:7822:3886#0

15. What is the start position of this read's mate on the genome?

```
(base) [root@522cc4c0d20e proj2]# samtools view athal_wu_0_A.bam | head -1
GAII05_0002:1:113:7822:3886#0   1187    Chr3    11699950        60      51M     =       11700332        433     AAAAAAAATGTAAAACTGCTAAATCTCTCCTCTCTAAAGAACTC
GTCCCCG CCCCCCBBBCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCBAAB??@ACBBCCCD     PQ:i:21 SM:i:37 UQ:i:0  MQ:i:37 XQ:i:0  RG:Z:H100223_GAII05_0002
```

a.
b. Chr3:11700332

Questions 16 - 20:
Using BEDtools, examine how many of the alignments at point 2 in the specified range overlap exons at the locus of interest. Use the BEDtools '-wo' option to only report non-zero overlaps. The list of exons is given in the included 'athal_wu_0_A_annot.gtf' GTF file.

16. How many overlaps (each overlap is reported on one line) are reported?

```
(base) [root@522cc4c0d20e proj2]# bedtools intersect -abam data.sorted.bam -b athal_wu_0_A_annot.gtf -bed > overlap.bed
(base) [root@522cc4c0d20e proj2]# wc -l overlap.bed
3101 overlap.bed
```

a.
b. **3101**

17. How many of these are 10 bases or longer?
    a. Size of overlap in column 22

```
3101 overlaps.bed
(base) [root@522cc4c0d20e proj2]# cut -f22 overlaps.bed | sort -nrk1 | grep -n "^9" | head -1
2900:9
```

b.
c. 2900 -1 = **2899**

18. How many alignments overlap the annotations?
    a. Alignment info in columns 1-12, minimum 1-5

```
2900:9
(base) [root@522cc4c0d20e proj2]# cut -f1-5 overlaps.bed | sort -u | wc -l
3101
```

    b.
    c. **3101**
19. Conversely, how many exons have reads mapped to them?
    a. Exon info in columns 13 -21

```
(base) [root@522cc4c0d20e proj2]# cut -f13-21 overlaps.bed | sort -u | wc -l
21
```

    b.
    c. **21**
20. If you were to convert the transcript annotations in the file "athal_wu_0_A_annot.gtf" into BED format, how many BED records would be generated?
    a. Number of transcripts in column 9

```
(base) [root@522cc4c0d20e proj2]# cut -f9 athal_wu_0_A_annot.gtf | cut -d ' ' -f1,2 | sort -u | wc -l
4
```

    b.
    c. **4**