

Day3_Mapping_Teaching_CORRECTED

October 5, 2023

1 DAY3 - MAPPING PRACTICE

In this practice, we are going to map short reads against a reference. To know, how well do the reads align back to the reference, we use bwa-mem2 and samtools to assess the basic alignment statistics.

In this exercise, we will use : * the *Bathycoccus prasinus* genome - File : GCA_002220235.1_ASM222023v1_genomic.fna * the **ILLUMINA READS** of the 1613 sample - Files 1613_R1.fastq.gz, 1613_R2.fastq.gz

The tool bwa needs 2 steps: - **Reference indexing**: `bwa-mem2 index reference` - **Mapping in itself**: `bwa-mem2 mem -R READGROUP [options] reference fastq1 fastq2 > out.sam`

1.1 Reference indexation - bwa_mem2 index

Before mapping, we need index reference file!

Go into REF directory and list the content of the directory

```
[ ]: # PUT COMMAND
```

```
[4]: cd /home/jovyan/SV_DATA/REF/  
pwd  
ls -lrt
```

```
/home/jovyan/SV_DATA/REF  
total 561996  
-rw-r--r-- 1 jovyan users 380710511 Oct 31 2011 all.chrs.con  
-rw-r--r-- 1 jovyan users 81498659 Feb 7 2012 all.gff3  
-rw----- 1 jovyan users 15263945 Sep 23 01:29  
GCA_002220235.1_ASM222023v1_genomic.fna  
-rw----- 1 jovyan users 5886829 Sep 23 15:25  
GCA_002220235.1_ASM222023v1_genomic.gff  
-rw-r--r-- 1 jovyan users 3768582 Sep 24 10:45  
GCA_002220235.1_ASM222023v1_genomic.fna.pac  
-rw-r--r-- 1 jovyan users 1587 Sep 24 10:45  
GCA_002220235.1_ASM222023v1_genomic.fna.ann  
-rw-r--r-- 1 jovyan users 308 Sep 24 10:45  
GCA_002220235.1_ASM222023v1_genomic.fna.amb  
-rw-r--r-- 1 jovyan users 30148640 Sep 24 10:45
```

```
GCA_002220235.1_ASM222023v1_genomic.fna.0123
-rw-r--r-- 1 jovyan users 48991633 Sep 24 10:45
GCA_002220235.1_ASM222023v1_genomic.fna.bwt.2bit.64
-rw-r--r-- 1 jovyan users 681 Sep 25 19:37
GCA_002220235.1_ASM222023v1_genomic.fna.fai
-rw-r--r-- 1 jovyan users 65 Oct 2 14:17 qtl.bed
-rw-r--r-- 1 jovyan users 594953 Oct 3 13:16 qtl.fasta
-rw-r--r-- 1 jovyan users 604866 Oct 3 13:18 qtl.formatseqtk.fasta
-rw-r--r-- 1 jovyan users 353 Oct 3 13:19 all.chrs.con.fai
-rw-r--r-- 1 jovyan users 7957814 Oct 3 13:23 all_genes.gff3
-rw-r--r-- 1 jovyan users 14204 Oct 3 13:27 qtl.intesect.genes.bed
```

Print the first lines of the file that contains the genome sequence (fasta format)

```
[ ]: # PUT COMMAND
```

```
[5]: head /home/jovyan/SV_DATA/REF/GCA_002220235.1_ASM222023v1_genomic.fna -n 5
```

```
>F0082278.1 Bathycoccus prasinus genomic : Chromosome_1
TTGCATAGTTTCTAGGGTCTGGGATTTTTGGGATTTTCCGAAAACCTCTTCTCTAACTGGACGgtctcggcgcgac
gagacTCCCGGTTGGGAATGTTCTTTTGGCGTGCAATCATATCCTGCTCCGGATACCGAATCGTGCGCTGCGAAATCGT
CTCCGCACTTGAAGGCTGTTTCTCACCATTACTATTATTAACGCTATCTCGAACTTCCTGCTTCGGTGCCGCCGGACCTA
CTCTGTTATTTCCCACGAGCTGTTgtttctcgctctctttgcTACTTTTACTAGATCACTCGTCGCGTCGAGAACATGC
```

Before mapping we need index reference file! Check bwa-mem2 index command line.

```
[6]: bwa-mem2
```

Usage: bwa-mem2 <command> <arguments>

Commands:

index	create index
mem	alignment
version	print version number

```
[7]: #PUT COMMAND TO GET HELP FOR bwa-mem2 index
bwa-mem2 index
```

Usage: bwa index [-p prefix] <in.fasta>

```
[ ]: # PUT COMMAND TO CREATE INDEX
```

```
[8]: bwa-mem2 index GCA_002220235.1_ASM222023v1_genomic.fna
```

```
[bwa_index] Pack FASTA... 0.06 sec
init ticks = 1077859628
ref seq len = 30148640
```

```

binary seq ticks = 553871378
build index ticks = 12798787546
ref_seq_len = 30148640
count = 0, 7834025, 15074320, 22314615, 30148640
BWT[28778507] = 4
CP_SHIFT = 5, CP_MASK = 31
sizeof CP_OCC = 64
max_occ_ind = 942145
ref_seq_len = 30148640
count = 0, 7834025, 15074320, 22314615, 30148640
BWT[28778507] = 4
CP_SHIFT = 6, CP_MASK = 63
sizeof CP_OCC = 64
max_occ_ind = 471072

```

Check that the indexes have been created

[9]: `ls -lrt`

```

total 896904
-rw-r--r-- 1 jovyan users 380710511 Oct 31 2011 all.chrs.con
-rw-r--r-- 1 jovyan users 81498659 Feb 7 2012 all.gff3
-rw----- 1 jovyan users 15263945 Sep 23 01:29
GCA_002220235.1_ASM222023v1_genomic.fna
-rw----- 1 jovyan users 5886829 Sep 23 15:25
GCA_002220235.1_ASM222023v1_genomic.gff
-rw-r--r-- 1 jovyan users 681 Sep 25 19:37
GCA_002220235.1_ASM222023v1_genomic.fna.fai
-rw-r--r-- 1 jovyan users 65 Oct 2 14:17 qtl.bed
-rw-r--r-- 1 jovyan users 594953 Oct 3 13:16 qtl.fasta
-rw-r--r-- 1 jovyan users 604866 Oct 3 13:18 qtl.formatseqtk.fasta
-rw-r--r-- 1 jovyan users 353 Oct 3 13:19 all.chrs.con.fai
-rw-r--r-- 1 jovyan users 7957814 Oct 3 13:23 all_genes.gff3
-rw-r--r-- 1 jovyan users 14204 Oct 3 13:27 qtl.intesect.genes.bed
-rw-r--r-- 1 jovyan users 3768582 Oct 3 13:47
GCA_002220235.1_ASM222023v1_genomic.fna.pac
-rw-r--r-- 1 jovyan users 1587 Oct 3 13:47
GCA_002220235.1_ASM222023v1_genomic.fna.ann
-rw-r--r-- 1 jovyan users 308 Oct 3 13:47
GCA_002220235.1_ASM222023v1_genomic.fna.amb
-rw-r--r-- 1 jovyan users 30148640 Oct 3 13:47
GCA_002220235.1_ASM222023v1_genomic.fna.0123
-rw-r--r-- 1 jovyan users 211040597 Oct 3 13:47
GCA_002220235.1_ASM222023v1_genomic.fna.bwt.8bit.32
-rw-r--r-- 1 jovyan users 180891925 Oct 3 13:47
GCA_002220235.1_ASM222023v1_genomic.fna.bwt.2bit.64

```

1.2 Run the mapping with bwa mem for only one sample (sample 1613)

Go into the directory SV_DATA

```
[ ]: # PUT COMMAND
```

```
[15]: pwd
      cd /home/jovyan/SV_DATA
      pwd
```

/home/jovyan/SV_DATA/results

/home/jovyan/SV_DATA

Create a directory RESULTS and go into the directory

```
[ ]: # PUT COMMAND
```

```
[16]: ### UPDATING
      mkdir -p RESULTS
```

```
[18]: cd RESULTS
      pwd
```

/home/jovyan/SV_DATA/RESULTS

```
[ ]: #PUT COMMAND TO GET HEP FOR bwa-mem2 mem
```

```
[13]: bwa-mem2 mem
```

Executing in AVX2 mode!!

Usage: bwa2 mem [options] <idxbase> <in1.fq> [in2.fq]

Options:

Algorithm options:

-o STR	Output SAM file name
-t INT	number of threads [1]
-k INT	minimum seed length [19]
-w INT	band width for banded alignment [100]
-d INT	off-diagonal X-dropoff [100]
-r FLOAT	look for internal seeds inside a seed longer than {-k} * FLOAT

[1.5]

-y INT	seed occurrence for the 3rd round seeding [20]
-c INT	skip seeds with more than INT occurrences [500]
-D FLOAT	drop chains shorter than FLOAT fraction of the longest overlapping chain [0.50]
-W INT	discard a chain if seeded bases shorter than INT [0]
-m INT	perform at most INT rounds of mate rescues for each read [50]
-S	skip mate rescue
-o	output file name missing
-P	skip pairing; mate rescue performed unless -S also in use

Scoring options:

-A INT score for a sequence match, which scales options -TdBOELU unless overridden [1]
-B INT penalty for a mismatch [4]
-O INT[,INT] gap open penalties for deletions and insertions [6,6]
-E INT[,INT] gap extension penalty; a gap of size k cost '{-O} + {-E}*k' [1,1]
-L INT[,INT] penalty for 5'- and 3'-end clipping [5,5]
-U INT penalty for an unpaired read pair [17]

Input/output options:

-p smart pairing (ignoring in2.fq)
-R STR read group header line such as '@RG\tID:foo\tSM:bar' [null]
-H STR/FILE insert STR to header if it starts with @; or insert lines in FILE [null]
-j treat ALT contigs as part of the primary assembly (i.e. ignore <idxbase>.alt file)
-v INT verbose level: 1=error, 2=warning, 3=message, 4+=debugging [3]
-T INT minimum score to output [30]
-h INT[,INT] if there are <INT hits with score >80% of the max score, output all in XA [5,200]
-a output all alignments for SE or unpaired PE
-C append FASTA/FASTQ comment to SAM output
-V output the reference FASTA header in the XR tag
-Y use soft clipping for supplementary alignments
-M mark shorter split hits as secondary
-I FLOAT[,FLOAT[,INT[,INT]]]
specify the mean, standard deviation (10% of the mean if absent), max
(4 sigma from the mean if absent) and min of the insert size distribution.

FR orientation only. [inferred]

Note: Please read the man page for detailed description of the command line and options.

No. of OMP threads: 0

Processor is running @2295.132606 MHz

Runtime profile:

Time taken for main_mem function: 0.00 Sec

IO times (sec) :

Reading IO time (reads) avg: 0.00, (0.00, 0.00)

Writing IO time (SAM) avg: 0.00, (0.00, 0.00)

Reading IO time (Reference Genome) avg: 0.00, (0.00, 0.00)

Index read time avg: 0.00, (0.00, 0.00)

Overall time (sec) (Excluding Index reading time):

PROCESS() (Total compute time + (read + SAM) IO time) : 0.00

MEM_PROCESS_SEQ() (Total compute time (Kernel + SAM)), avg: 0.00, (0.00,

0.00)

SAM Processing time (sec):
--WORKER_SAM avg: 0.00, (0.00, 0.00)

Kernels' compute time (sec):
Total kernel (smem+sal+bsw) time avg: 0.00, (0.00, 0.00)
SMEM compute avg: -nan, (0.00, 435704.67)
SAL compute avg: -nan, (0.00, 435704.67)
BSW time, avg: -nan, (0.00, 435704.67)

Total allocs: 0 = 0 out total requests: 0, Rate: -nan

Important parameter settings:

BATCH_SIZE: 512
MAX_SEQ_LEN_REF: 256
MAX_SEQ_LEN_QER: 128
MAX_SEQ_LEN8: 128
SEEDS_PER_READ: 500
SIMD_WIDTH8 X: 32
SIMD_WIDTH16 X: 16
AVG_SEEDS_PER_READ: 64

[]: # PUT COMMAND

[28]: bwa-mem2 mem -R "@RG\tID:1613\tSM:1613" -M -t 8 ../REF/GCA_002220235.
↪1_ASM222023v1_genomic.fna ../SHORT_READS/1613_R1.fastq.gz ../SHORT_READS/
↪1613_R2.fastq.gz > 1613.sam

Executing in AVX2 mode!!

Ref file: ../REF/GCA_002220235.1_ASM222023v1_genomic.fna

Entering FMI_search

reference seq len = 30148641

count

0,	1
1,	7834026
2,	15074321
3,	22314616
4,	30148641

Reading other elements of the index from files

../REF/GCA_002220235.1_ASM222023v1_genomic.fna

prefix: ../REF/GCA_002220235.1_ASM222023v1_genomic.fna

[M::bwa_idx_load_ele] read 0 ALT contigs

Done reading Index!!

Reading reference genome..

Binary seq file = ../REF/GCA_002220235.1_ASM222023v1_genomic.fna.0123
Reference genome size: 30148640 bp
Done readng reference genome !!

[0000] 1: Calling process()

Threads used (compute): 8

Info: projected #read in a task: 529811

Memory pre-allocation for chaining: 1114.7223 MB

Memory pre-allocation for BSW: 1916.9362 MB

Memory pre-allocation for BWT: 618.5134 MB

No. of pipeline threads: 2

[0000] read_chunk: 80000000, work_chunk_size: 80000217, nseq: 320478

[0000][M::kt_pipeline] read 320478 sequences (80000217 bp)...

[0000] 2. Calling mem_process_seqs..., task: 0

[0000] 3. Calling kt_for - worker_bwt

[0000] read_chunk: 80000000, work_chunk_size: 80000178, nseq: 320478

[0000][M::kt_pipeline] read 320478 sequences (80000178 bp)...

[0000] 3. Calling kt_for - worker_aln

Inferring insert size distribution from data, l_pac: 15074320, n: 320478

[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (6, 146360, 9, 7)

[M::mem_pestat] skip orientation FF as there are not enough pairs

[M::mem_pestat] analyzing insert size distribution for orientation FR...

[M::mem_pestat] (25, 50, 75) percentile: (228, 278, 344)

[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 576)

[M::mem_pestat] mean and std.dev: (290.09, 87.34)

[M::mem_pestat] low and high boundaries for proper pairs: (1, 692)

[M::mem_pestat] skip orientation RF as there are not enough pairs

[M::mem_pestat] skip orientation RR as there are not enough pairs

[0000] 10. Calling kt_for - worker_sam

[0000][M::mem_process_seqs] Processed 320478 reads in 47.920 CPU sec,
6.018 real sec

[0000] 2. Calling mem_process_seqs..., task: 1

[0000] 3. Calling kt_for - worker_bwt

[0000] read_chunk: 80000000, work_chunk_size: 80000142, nseq: 320476

[0000][M::kt_pipeline] read 320476 sequences (80000142 bp)...

[0000] 3. Calling kt_for - worker_aln

Inferring insert size distribution from data, l_pac: 15074320, n: 320478

[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (7, 146471, 14,
7)

[M::mem_pestat] skip orientation FF as there are not enough pairs

[M::mem_pestat] analyzing insert size distribution for orientation FR...

[M::mem_pestat] (25, 50, 75) percentile: (227, 277, 343)

[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 575)

[M::mem_pestat] mean and std.dev: (289.75, 87.63)

[M::mem_pestat] low and high boundaries for proper pairs: (1, 691)

```

[M::mem_pestat] analyzing insert size distribution for orientation RF...
[M::mem_pestat] (25, 50, 75) percentile: (60, 334, 2198)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1,
6474)
[M::mem_pestat] mean and std.dev: (1083.00, 1224.54)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 8612)
[M::mem_pestat] skip orientation RR as there are not enough pairs
[M::mem_pestat] skip orientation RF
[0000] 10. Calling kt_for - worker_sam
      [0000][ M::mem_process_seqs] Processed 320478 reads in 45.630 CPU sec,
5.652 real sec
[0000] 2. Calling mem_process_seqs..., task: 2
[0000] 3. Calling kt_for - worker_bwt
[0000] read_chunk: 80000000, work_chunk_size: 80000457, nseq: 320478
      [0000][ M::kt_pipeline] read 320478 sequences (80000457 bp)...
[0000] 3. Calling kt_for - worker_aln
Inferring insert size distribution from data, l_pac: 15074320, n: 320476
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (13, 146615, 11,
9)
[M::mem_pestat] analyzing insert size distribution for orientation FF...
[M::mem_pestat] (25, 50, 75) percentile: (3706, 4374, 7453)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1,
14947)
[M::mem_pestat] mean and std.dev: (4598.31, 2463.35)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 18694)
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (227, 277, 343)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 575)
[M::mem_pestat] mean and std.dev: (289.61, 87.35)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 691)
[M::mem_pestat] analyzing insert size distribution for orientation RF...
[M::mem_pestat] (25, 50, 75) percentile: (156, 217, 2198)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1,
6282)
[M::mem_pestat] mean and std.dev: (1123.36, 1223.29)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 8324)
[M::mem_pestat] skip orientation RR as there are not enough pairs
[M::mem_pestat] skip orientation FF
[M::mem_pestat] skip orientation RF
[0000] 10. Calling kt_for - worker_sam
      [0000][ M::mem_process_seqs] Processed 320476 reads in 46.026 CPU sec,
5.680 real sec
[0000] 2. Calling mem_process_seqs..., task: 3
[0000] 3. Calling kt_for - worker_bwt
[0000] read_chunk: 80000000, work_chunk_size: 47353187, nseq: 189696
      [0000][ M::kt_pipeline] read 189696 sequences (47353187 bp)...
[0000] 3. Calling kt_for - worker_aln
Inferring insert size distribution from data, l_pac: 15074320, n: 320478

```

```

[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (6, 146206, 5, 5)
[M::mem_pestat] skip orientation FF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (227, 276, 342)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 572)
[M::mem_pestat] mean and std.dev: (288.48, 86.88)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 687)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] skip orientation RR as there are not enough pairs
[0000] 10. Calling kt_for - worker_sam
      [0000] [ M::mem_process_seqs] Processed 320478 reads in 45.713 CPU sec,
5.704 real sec
[0000] 2. Calling mem_process_seqs..., task: 4
[0000] 3. Calling kt_for - worker_bwt
[0000] read_chunk: 80000000, work_chunk_size: 0, nseq: 0
[0000] 3. Calling kt_for - worker_aln
Inferring insert size distribution from data, l_pac: 15074320, n: 189696
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (2, 86576, 7, 5)
[M::mem_pestat] skip orientation FF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (227, 277, 343)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (1, 575)
[M::mem_pestat] mean and std.dev: (289.32, 87.35)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 691)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] skip orientation RR as there are not enough pairs
[0000] 10. Calling kt_for - worker_sam
      [0000] [ M::mem_process_seqs] Processed 189696 reads in 27.671 CPU sec,
3.486 real sec
[0000] read_chunk: 80000000, work_chunk_size: 0, nseq: 0
[0000] Computation ends..
No. of OMP threads: 8
Processor is running @2295.048362 MHz
Runtime profile:

```

```

      Time taken for main_mem function: 28.08 Sec

```

```

      IO times (sec) :

```

```

      Reading IO time (reads) avg: 4.88, (4.88, 4.88)

```

```

      Writing IO time (SAM) avg: 1.94, (1.94, 1.94)

```

```

      Reading IO time (Reference Genome) avg: 0.02, (0.02, 0.02)

```

```

      Index read time avg: 0.18, (0.18, 0.18)

```

```

      Overall time (sec) (Excluding Index reading time):

```

```

      PROCESS() (Total compute time + (read + SAM) IO time) : 27.88

```

```

      MEM_PROCESS_SEQ() (Total compute time (Kernel + SAM)), avg: 26.53,
(26.53, 26.53)

```

```
SAM Processing time (sec):
--WORKER_SAM avg: 3.84, (3.84, 3.84)
```

```
Kernels' compute time (sec):
Total kernel (smem+sal+bsw) time avg: 22.38, (22.38, 22.38)
    SMEM compute avg: 8.15, (8.22, 8.12)
    SAL compute avg: 0.85, (0.88, 0.81)
    BSW time, avg: 12.87, (12.93, 12.77)
```

```
Total allocs: 1941303 = 1941303 out total requests: 2368737, Rate: 0.82
```

Important parameter settings:

```
BATCH_SIZE: 512
MAX_SEQ_LEN_REF: 256
MAX_SEQ_LEN_QER: 128
MAX_SEQ_LEN8: 128
SEEDS_PER_READ: 500
SIMD_WIDTH8 X: 32
SIMD_WIDTH16 X: 16
AVG_SEEDS_PER_READ: 64
```

Check that the file .sam have been created by bwa mem

```
[ ]: # PUT COMMAND
```

```
[31]: ls -lh
```

```
total 902M
-rw-r--r-- 1 jovyan users 902M Oct  3 14:01 1613.sam
```

Display the first and the end of the sam file just created

```
[ ]: # PUT COMMANDS
```

```
[37]: head 1613.sam -n 30
```

```
@SQ      SN:F0082278.1    LN:1352724
@SQ      SN:F0082277.1    LN:1122692
@SQ      SN:F0082276.1    LN:1091008
@SQ      SN:F0082275.1    LN:1037991
@SQ      SN:F0082274.1    LN:1019276
@SQ      SN:F0082273.1    LN:989707
@SQ      SN:F0082272.1    LN:955652
@SQ      SN:F0082271.1    LN:937610
@SQ      SN:F0082270.1    LN:895536
@SQ      SN:F0082269.1    LN:794368
@SQ      SN:F0082268.1    LN:741603
@SQ      SN:F0082267.1    LN:712459
@SQ      SN:F0082266.1    LN:708035
```

```

@SQ      SN:F0082265.1   LN:663424
@SQ      SN:F0082264.1   LN:519835
@SQ      SN:F0082263.1   LN:494108
@SQ      SN:F0082262.1   LN:465570
@SQ      SN:F0082261.1   LN:310170
@SQ      SN:F0082260.1   LN:146238
@SQ      SN:F0082258.2   LN:43614
@SQ      SN:F0082259.2   LN:72700
@RG      ID:1613 SM:1613
@PG      ID:bwa PN:bwa VN:2.0pre2 CL:bwa-mem2 mem -R @RG\tID:1613\tSM:1613
-M -t 8 ../REF/GCA_002220235.1_ASM222023v1_genomic.fna
../SHORT_READS/1613_R1.fastq.gz ../SHORT_READS/1613_R2.fastq.gz
M02713:279:GW200319000:1:2106:15454:1334      83      F0082272.1      908208
60      18S232M =      908208 -232      CGTGTGCTCTTCCGATCTAACCGAGCGTGCCTTTCTTCTT
TTTTCTGTAAAAGGAAGAACTAGTAGTAGTCGTAGCGTCGCCGTCTTCGTCTTCCATCGTCAAAGCTTCTTTCTCGCTC
TCTCCACGGTGCTCGCTCGAACCCACCGACGTTATCCCTTGATTATCGCTCGTTTCCGTCTCCCTCGTTTTTGTTC
ACGACGGTCCATCCGTCGTCTTCTCTCTCGCGCGTTTCCGTTTTTC      -FFFF..EFFFDEFFFDFFB;FF
FFFFEA/FFEFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFFD?FFFFFFFFFFFFFFFFBFDFFFFFFFFFFFF
FFFBGEGGAGGGGGEGGEDCE;AEHGD?F@C/G?HCGCFDE<GGGFGFHFFFHFAFD22DFGGHGGGGGGHEFGGGGFGE
?GECFHHG1EFGHHGGEFEHGHCFGEFGGGGGEHHHGHGCHCGGEAECEGFFBBA?BAFAABBA      NM:i:1
MD:Z:71C160      AS:i:227      XS:i:19 RG:Z:1613
M02713:279:GW200319000:1:2106:15454:1334      163      F0082272.1      908208
60      232M18S =      908208 232      AACCGAGCGTGCCTTTCTTCTTTTTCTGTAAAAGGAAGA
ACTAGTAGTAGTCGTAGCGTCGCCGTCTTCGTCTTCCATCGTCAAAGCTTCTTTCTCGCTCTCTCCACGGTGCTCGCTC
GAACCCACCGACGGTTATCCCTTGATTATCGCTCGTTTCCGTCTCCCTCGTTTTTGTTCACGACGGTCCATCCGCCG
TCTTCTCTTCTCTCGCGCGTTTCCGTTTTTTCAGCTCGGAAGAGCGTCGT      1AAAA1@DDAAAGGE1FGGGGGHCH
HGHGHHFHG2F/OGGCCHFHFDGFGGHHHGEFGGGCE?/EEG@OFGGHCA>FHDGHG>GHGH1CGHFHHEFFF>FE>/EH
GHEHHGGEHGD@FH/@FGGGGGCC?@C@FECFHBGBH/DHFGHHE?AGCAE.CH0.9../9/BFGC.=EA9CFFBF0
;.@9@AB@BFBAB-;@-@9A9/9FF9/B/BFFF@C@=9@BBABFFFAFBF//9-9A---;B-AB>@      NM:i:2
MD:Z:71C125T34      AS:i:222      XS:i:19 RG:Z:1613
M02713:279:GW200319000:1:2106:15870:1335      99      F0082278.1      612747
60      250M =      612783 286      GTCTGGATCGAGTGCCTTCGGTAAACGGCGGGAATTTCGA
GGGCGTGGACATCCCATCGATTCAAACAGTCGCCGAGCGATGGGCATCGAGAAACGGCTGCCAAACGGAAGAATCGGTGG
TTGTTGATGGAGCGCTGAATTTGCGCACGAAGCTTTTCGCGAATGCGCCGGATACTTCCTCAGAAACGTGGACGAATGGT
ACGTGCCAAGAAGGAACAGTTGTTGAGAATTGGGTAATCGACGGGATGAA      A>AABFFDC?AA44AGFGGGGFHD
F5AAEEGGGAAGHFFGHGCDDEE/EAG3B@?G3@31?/GEFGHHHB4FEEECGCD/CD/B?GHHC0/<FADGBEACGGGACG
FHGGDCF<GGOGG<ED.CFAC.OGC:CGB?-EGAFGGG00B?BGGAEGBFFFFB?@BBBF/ADA--
@DFBBBFBFFFFBB/9AAAA?/--BAFFFF/.;FFAFFEFFF?FB9;9:BB9AFFFAFBFB/9.99BBEFBBB-.9A/F
NM:i:2 MD:Z:61C83A104      AS:i:240      XS:i:0 RG:Z:1613
M02713:279:GW200319000:1:2106:15870:1335      147      F0082278.1      612783
60      250M =      612747 -286      TCGAGGGCTTGGACATCCCATCGATTCAAACAGTCGCCGA
GCGATGGGCATCGAGAAACGGCTGCCAAACGGAAGAATCGGTGGTTGTTGATGGAGCGCTGAATTTGCGCACGAAGCTTT
TCGCGAATGCGCCGATACTTCCTCAGAAACGTGGACGAATGGTACGTGCCAAGAAGGAACAGTTGTTGAGAATTGGGTA
ATCGACGGGATGAATACCGATACGGCGGTTGCCAACTCGTTACGGATAC      .@9.9AAB/;/9AB.?DFAFFF;
//BF/BFA=@-
.A>AA.DFFFF/?FBFA9FBB.DFE.DFB;.FFGDFGFFFGBADFGE?EEFFFGF;/.DFGAOCCOHD?CCDCC<@DGG
OHGFD@</FGGCGCD@?GHFGGCOB3??B?BGFGCE?/GGHEFGFHGHGHGFBFEHHGDGHHGFHHHGHGHGHFHEH
HFHFGGGE1EFF3BHGFGEHFFGHFFGE2CEE?CGGGGFEECGFABAAFA3AAA      NM:i:3

```



```

FHFEGE?FHHHHHHHHFHHHGFGFAEHHHHHHHFGHEH2EHFBHDGHHGFFHHHFFHHFHHHHGGHGFH1FG1GEGGGGH
FHHFHFHHFF/GEGGGGGGHHGHHGGA@-
EGCC<GFHHHGHGHHGHHCCCEHECCCGCFHHDCAAFF? .A??B@GB?>;B;B?;BBB>BB=AB@EB=9=BFF?FEDA
--DD?9.-B.;FFBEDDB=.;ABF==FFEAFFF.ABB      NM:i:1  MD:Z:76C160      AS:i:232
XS:i:0  RG:Z:1613
M02713:279:GW200319000:1:1110:16398:29280      147      F0082262.1      100253
60      14S236M =      100253  -236      CGCTCTTCCGATCTGCAATCGTCGCCATGGAGAAGTATTG
CGAAGAGGTTGAAAACTTCAACCAAACTGGTGGCAGTTGATGACTGAAACTGCACGTACGAAGAGAAGGCCAAGATGT
TATCGATGGCGGACGAATCCTCGGAGGAGTATTATTACAGGTTGGGATGCGGTTGCTCGTACCTCCGACGGGACGGGG
ACGACGACGTCGTCGTCGTCCTTCGAGAGGGAGGAATACGGACGTTCCG
?FFFFEFFFFFFFFFEE;BBBBFFFFBFEEFFFFFFFFFFFFFFFFBFFFFFFFFBFFFFEE-FFFFFFFFFFFFFFFFFG
GFEGFGGGGGFGGGGGGGFGGGGGGGGFFCCOHHHG. CDFGGGCGFGEHHFGCGGGFHHHGHGHHGDHHGHGHHH
GGHGHGHHGGCCGFHD@EGGHGECEE?GGGGGGGGGHGFGGGFGE?GGGGGGCGGHEAGGGHHHGHGE4DFFGGEF>2
AAA@BABBA      NM:i:1  MD:Z:76C159      AS:i:231      XS:i:0  RG:Z:1613
M02713:279:GW200319000:1:1110:16195:29320      99      F0082266.1      424313
60      250M =      424557  493      TGTATCACATATATGAATACACAAACAAACAAACAAACGT
GTGCTGAAAAGTGCAGATTTTTTCCCGCCAAAATTGTTCTCCATTGTAATGGAGCACAAAAGAGTTGTGTTTATTAATG
AATAACAACAACACCAAAAACAAAGGTTGCTCGGAAACGAGACGAACGAGTGTGCTGCGACTGCGACTGCTGCGCC
TGTTCCTCTGAAGGTGGTGTCTGTTCTCGAAGAAGAGAAGGAGGAGAAGA      BABBBBD5DBFBFGFBGGGGGGHG
GHHHAGHDHHHFFHEFFHGGHHFHHHFFHHHBAFGHHHGGHHG?EEHHHHHHHHHHHGHGHHHHHHCHHHHGHGBFH
4FHFHHHHHHHHHHHHHHHHHHHHHGHGFF?HHHFGHHHHHFFHFDFFDCCGGHHGFCFGGGGGGGDGAFFCGHHGC
GGGGDGGGGGGFFGAGGEGGFGFFFGGGGGFBFFFFE/.;AAEF. :D;BFE.;BFF?DDFFFF      NM:i:0
MD:Z:250      AS:i:250      XS:i:0  RG:Z:1613
M02713:279:GW200319000:1:1110:16195:29320      147      F0082266.1      424557
60      249M =      424313  -493      AGAAGAATGATGTTGACGATGATGATGAAATCATCATCAT
CGAAGAATCTTTAAAGAATAATAATAAGAAGAAGCATTCAAAGACACCGTCAAAAACGACAACGACAACATCAACGAAAA
GGAAGAAGAACGATACTACCGATAATAAGAAAAGCAGTGATAAAAATGATCCGTATTCGTTGAAAACGCGAAAGAGTTG
TTCGGAAGTGGACGAGGAAATTTTGGTCGAGGATGATGATTATGAAG      FBFFCBFFGBBGGGGGGGGGGGFG
FGGGGGGFFGFFFFGFGHHHFFHHHHFHHEHHGOHHGFHHHHFHFBHGBD<DGFGFFGGHEGGHCGGGGGGGHGGGGGH
HHHGEHHHGC GFHHHHHHHHHHHGHGHHHGGHGGGFFHHHGGHGHHHHHHHHHHFFHHHFFHHHFGGGEAHG/?>EHEE2F
GGGGGGHFFHFGAHGGGGGFFHHHCHGGGHFFGHFFHHGCEEGCGGGGF3GGFFFFDB3FA@AAA      NM:i:2
MD:Z:19C222G6  AS:i:239      XS:i:0  RG:Z:1613

```

Sort sam file and convert it into a bam file (compressed format) - `samtools sort`

```
[ ]: # PUT COMMAND
```

```
[40]: samtools sort -@ 8 -o 1613.sorted.bam 1613.sam
```

[bam_sort_core] merging from 0 files and 8 in-memory blocks...

Check that the bam file have been created

- Have a look at the filesize of the sam and bam files.
- Remove the sam file

```
[ ]: # PUT COMMAND
```

```
[ ]: # PUT COMMAND
```

```
[41]: ls -lh
      rm 1613.sam
```

```
total 1.1G
-rw-r--r-- 1 jovyan users 902M Oct  3 14:01 1613.sam
-rw-r--r-- 1 jovyan users 167M Oct  3 14:03 1613.sorted.bam
```

Calculate some stats about mapping with samtools flagstat

```
[ ]: # PUT COMMAND
```

```
[42]: samtools flagstat 1613.sorted.bam
```

```
1492864 + 0 in total (QC-passed reads + QC-failed reads)
1471606 + 0 primary
21258 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
1439155 + 0 mapped (96.40% : N/A)
1417897 + 0 primary mapped (96.35% : N/A)
1471606 + 0 paired in sequencing
735803 + 0 read1
735803 + 0 read2
1392750 + 0 properly paired (94.64% : N/A)
1413380 + 0 with itself and mate mapped
4517 + 0 singletons (0.31% : N/A)
16728 + 0 with mate mapped to a different chr
14781 + 0 with mate mapped to a different chr (mapQ>=5)
```

Calculate some stats about mapping with samtools flagstat and save results in the file 1613.flagstat

```
[ ]: # PUT COMMAND
```

```
[43]: samtools flagstat 1613.sorted.bam >1613.flagstat
```

```
[44]: ls -lrt
```

```
total 170276
-rw-r--r-- 1 jovyan users 174358429 Oct  3 14:03 1613.sorted.bam
-rw-r--r-- 1 jovyan users      521 Oct  3 14:06 1613.flagstat
```

Display the content of the flagstat file

```
[45]: cat 1613.flagstat
```

```
1492864 + 0 in total (QC-passed reads + QC-failed reads)
1471606 + 0 primary
```

```

21258 + 0 secondary
0 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
1439155 + 0 mapped (96.40% : N/A)
1417897 + 0 primary mapped (96.35% : N/A)
1471606 + 0 paired in sequencing
735803 + 0 read1
735803 + 0 read2
1392750 + 0 properly paired (94.64% : N/A)
1413380 + 0 with itself and mate mapped
4517 + 0 singletons (0.31% : N/A)
16728 + 0 with mate mapped to a different chr
14781 + 0 with mate mapped to a different chr (mapQ>=5)

```

1.3 Generate a bam file that contains only the reads correctly paired mapped samtools view

<https://broadinstitute.github.io/picard/explain-flags.html>

```
[ ]: # PUT COMMAND
```

```
[46]: samtools view -bh -@4 -f 0x02 -o 1613.onlymappedpaired.bam 1613.sorted.bam
```

```
[47]: ls -lhrt
```

```

total 321M
-rw-r--r-- 1 jovyan users 167M Oct  3 14:03 1613.sorted.bam
-rw-r--r-- 1 jovyan users  521 Oct  3 14:06 1613.flagstat
-rw-r--r-- 1 jovyan users 154M Oct  3 14:07 1613.onlymappedpaired.bam

```

1.4 Indexing the last bam file generated that will be used to detect SNPs

```
[ ]: # PUT COMMAND
```

```
[48]: samtools index 1613.onlymappedpaired.bam
ls -lhrt
```

```

total 321M
-rw-r--r-- 1 jovyan users 167M Oct  3 14:03 1613.sorted.bam
-rw-r--r-- 1 jovyan users  521 Oct  3 14:06 1613.flagstat
-rw-r--r-- 1 jovyan users 154M Oct  3 14:07 1613.onlymappedpaired.bam
-rw-r--r-- 1 jovyan users  46K Oct  3 14:09 1613.onlymappedpaired.bam.bai

```

1.5 Let's mapping an other sample (1868) to reduce steps and temporary bam files using the pipe |

The pipe is a form of redirection in Linux used to connect the STDOUT of one command into the STDIN of a second command.

1- Generate the bam file 1868.sorted.bam Use a single command line to perform : * mapping with `bwa-mem2 mem` * sort the mapping results with `samtools sort` * and only the file `1868.sorted.bam` is generated

```
[ ]: # PUT COMMAND
```

```
[5]: bwa-mem2 mem -R "@RG\tID:1868\tSM:1868" -t 8 ../REF/GCA_002220235.
      ↪1_ASM222023v1_genomic.fna ../SHORT_READS/1868_R1.fastq.gz ../SHORT_READS/
      ↪1868_R2.fastq.gz | samtools sort -@ 8 -o 1868.sorted.bam
```

```
-----
Executing in AVX2 mode!!
-----
```

```
Ref file: ../REF/GCA_002220235.1_ASM222023v1_genomic.fna
```

```
Entering FMI_search
```

```
reference seq len = 30148641
```

```
count
```

```
0,      1
1,     7834026
2,     15074321
3,     22314616
4,     30148641
```

```
Reading other elements of the index from files
```

```
../REF/GCA_002220235.1_ASM222023v1_genomic.fna
```

```
prefix: ../REF/GCA_002220235.1_ASM222023v1_genomic.fna
```

```
[M::bwa_idx_load_ele] read 0 ALT contigs
```

```
Done reading Index!!
```

```
Reading reference genome..
```

```
Binary seq file = ../REF/GCA_002220235.1_ASM222023v1_genomic.fna.0123
```

```
Reference genome size: 30148640 bp
```

```
Done readng reference genome !!
```

```
[0000] 1: Calling process()
```

```
Threads used (compute): 8
```

```
Info: projected #read in a task: 529811
-----
```

```
Memory pre-allocation for chaining: 1114.7223 MB
```

```
Memory pre-allocation for BSW: 1916.9362 MB
```

```
Memory pre-allocation for BWT: 618.5134 MB
-----
```

```
No. of pipeline threads: 2
```

```
[0000] read_chunk: 80000000, work_chunk_size: 80000378, nseq: 320474
```

```
    [0000][ M::kt_pipeline] read 320474 sequences (80000378 bp)...
```

```
[0000] 2. Calling mem_process_seqs.., task: 0
```

```
[0000] 3. Calling kt_for - worker_bwt
```

```
[0000] read_chunk: 80000000, work_chunk_size: 80000317, nseq: 320476
```

```

[0000][ M::kt_pipeline] read 320476 sequences (80000317 bp)...
[0000] 3. Calling kt_for - worker_aln
Inferring insert size distribution from data, l_pac: 15074320, n: 320474
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (1, 140082, 4, 2)
[M::mem_pestat] skip orientation FF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (215, 256, 308)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (29,
494)
[M::mem_pestat] mean and std.dev: (264.14, 71.38)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 587)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] skip orientation RR as there are not enough pairs
[0000] 10. Calling kt_for - worker_sam
[0000][ M::mem_process_seqs] Processed 320474 reads in 38.813 CPU sec,
4.866 real sec
[0000] 2. Calling mem_process_seqs..., task: 1
[0000] 3. Calling kt_for - worker_bwt
[0000] read_chunk: 80000000, work_chunk_size: 80000454, nseq: 320474
[0000][ M::kt_pipeline] read 320474 sequences (80000454 bp)...
[0000] 3. Calling kt_for - worker_aln
Inferring insert size distribution from data, l_pac: 15074320, n: 320476
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (6, 139952, 4, 2)
[M::mem_pestat] skip orientation FF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (215, 256, 308)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (29,
494)
[M::mem_pestat] mean and std.dev: (264.15, 71.74)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 587)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] skip orientation RR as there are not enough pairs
[0000] 10. Calling kt_for - worker_sam
[0000][ M::mem_process_seqs] Processed 320476 reads in 38.959 CPU sec,
4.800 real sec
[0000] 2. Calling mem_process_seqs..., task: 2
[0000] 3. Calling kt_for - worker_bwt
[0000] read_chunk: 80000000, work_chunk_size: 80000276, nseq: 320476
[0000][ M::kt_pipeline] read 320476 sequences (80000276 bp)...
[0000] 3. Calling kt_for - worker_aln
Inferring insert size distribution from data, l_pac: 15074320, n: 320474
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (2, 139828, 5, 6)
[M::mem_pestat] skip orientation FF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (214, 255, 308)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (26,
496)
[M::mem_pestat] mean and std.dev: (263.87, 72.06)

```

```

[M::mem_pestat] low and high boundaries for proper pairs: (1, 590)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] skip orientation RR as there are not enough pairs
[0000] 10. Calling kt_for - worker_sam
      [0000][ M::mem_process_seqs] Processed 320474 reads in 38.401 CPU sec,
4.718 real sec
[0000] 2. Calling mem_process_seqs..., task: 3
[0000] 3. Calling kt_for - worker_bwt
[0000] read_chunk: 80000000, work_chunk_size: 24368032, nseq: 97618
      [0000][ M::kt_pipeline] read 97618 sequences (24368032 bp)...
[0000] 3. Calling kt_for - worker_aln
Inferring insert size distribution from data, l_pac: 15074320, n: 320476
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (2, 139969, 5, 2)
[M::mem_pestat] skip orientation FF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (215, 256, 307)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (31,
491)
[M::mem_pestat] mean and std.dev: (263.51, 71.18)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 583)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] skip orientation RR as there are not enough pairs
[0000] 10. Calling kt_for - worker_sam
      [0000][ M::mem_process_seqs] Processed 320476 reads in 37.521 CPU sec,
4.724 real sec
[0000] 2. Calling mem_process_seqs..., task: 4
[0000] 3. Calling kt_for - worker_bwt
[0000] read_chunk: 80000000, work_chunk_size: 0, nseq: 0
[0000] 3. Calling kt_for - worker_aln
Inferring insert size distribution from data, l_pac: 15074320, n: 97618
[M::mem_pestat] # candidate unique pairs for (FF, FR, RF, RR): (0, 42657, 1, 0)
[M::mem_pestat] skip orientation FF as there are not enough pairs
[M::mem_pestat] analyzing insert size distribution for orientation FR...
[M::mem_pestat] (25, 50, 75) percentile: (214, 256, 307)
[M::mem_pestat] low and high boundaries for computing mean and std.dev: (28,
493)
[M::mem_pestat] mean and std.dev: (263.48, 71.13)
[M::mem_pestat] low and high boundaries for proper pairs: (1, 586)
[M::mem_pestat] skip orientation RF as there are not enough pairs
[M::mem_pestat] skip orientation RR as there are not enough pairs
[0000] 10. Calling kt_for - worker_sam
      [0000][ M::mem_process_seqs] Processed 97618 reads in 11.981 CPU sec,
1.499 real sec
[0000] read_chunk: 80000000, work_chunk_size: 0, nseq: 0
[0000] Computation ends..
No. of OMP threads: 8
Processor is runnig @2295.055392 MHz
Runtime profile:

```

```

Time taken for main_mem function: 21.98 Sec

IO times (sec) :
Reading IO time (reads) avg: 4.64, (4.64, 4.64)
Writing IO time (SAM) avg: 1.86, (1.86, 1.86)
Reading IO time (Reference Genome) avg: 0.02, (0.02, 0.02)
Index read time avg: 0.16, (0.16, 0.16)

Overall time (sec) (Excluding Index reading time):
PROCESS() (Total compute time + (read + SAM) IO time) : 21.80
MEM_PROCESS_SEQ() (Total compute time (Kernel + SAM)), avg: 20.60,
(20.60, 20.60)

SAM Processing time (sec):
--WORKER_SAM avg: 2.51, (2.51, 2.51)

Kernels' compute time (sec):
Total kernel (smem+sal+bsw) time avg: 17.86, (17.86, 17.86)
    SMEM compute avg: 7.31, (7.37, 7.26)
    SAL compute avg: 0.73, (0.76, 0.69)
    BSW time, avg: 9.39, (9.45, 9.28)

Total allocs: 1640189 = 1640189 out total requests: 1975792, Rate: 0.83

Important parameter settings:
    BATCH_SIZE: 512
    MAX_SEQ_LEN_REF: 256
    MAX_SEQ_LEN_QER: 128
    MAX_SEQ_LEN8: 128
    SEEDS_PER_READ: 500
    SIMD_WIDTH8 X: 32
    SIMD_WIDTH16 X: 16
    AVG_SEEDS_PER_READ: 64
[bam_sort_core] merging from 0 files and 8 in-memory blocks...

```

```
[53]: ls -lrt
```

```

total 489304
-rw-r--r-- 1 jovyan users 174358429 Oct  3 14:03 1613.sorted.bam
-rw-r--r-- 1 jovyan users      521 Oct  3 14:06 1613.flagstat
-rw-r--r-- 1 jovyan users 161308863 Oct  3 14:07 1613.onlymappedpaired.bam
-rw-r--r-- 1 jovyan users   46104 Oct  3 14:09 1613.onlymappedpaired.bam.bai
-rw-r--r-- 1 jovyan users 165320351 Oct  3 14:28 1615.sorted.bam

```

Filtering with samtools view and create the index

```
[ ]: # PUT COMMAND
```

```
[6]: samtools view -bh -@8 -f 0x02 -o 1868.onlymappedpaired.bam 1868.sorted.bam
samtools index 1868.onlymappedpaired.bam
```

Getting some stats on mapping step - samtools flagstat We want the stat on the two bam files generated : * 1868.sorted.bam * 1868.onlymappedpaired.bam

```
[ ]: # PUT COMMAND
```

```
[7]: samtools flagstat 1868.sorted.bam >1868.flagstat
samtools flagstat 1868.onlymappedpaired.bam > 1868.onlymappedpaired.flagstat
```

```
[9]: ls -lht
```

```
total 660M
-rw-r--r-- 1 jovyan users 512 Oct 3 15:42 1868.onlymappedpaired.flagstat
-rw-r--r-- 1 jovyan users 519 Oct 3 15:42 1868.flagstat
-rw-r--r-- 1 jovyan users 46K Oct 3 15:41 1868.onlymappedpaired.bam.bai
-rw-r--r-- 1 jovyan users 137M Oct 3 15:41 1868.onlymappedpaired.bam
-rw-r--r-- 1 jovyan users 158M Oct 3 15:41 1868.sorted.bam
-rw-r--r-- 1 jovyan users 36M Oct 3 15:34 1616_1868.vcf
-rw-r--r-- 1 jovyan users 8.8M Oct 3 15:01 1616_1868.bcf
-rw-r--r-- 1 jovyan users 46K Oct 3 14:09 1613.onlymappedpaired.bam.bai
-rw-r--r-- 1 jovyan users 154M Oct 3 14:07 1613.onlymappedpaired.bam
-rw-r--r-- 1 jovyan users 521 Oct 3 14:06 1613.flagstat
-rw-r--r-- 1 jovyan users 167M Oct 3 14:03 1613.sorted.bam
```

Compare the two flagstat files. What do you think about the results ?

```
[10]: cat 1868*.flagstat
```

```
1393482 + 0 in total (QC-passed reads + QC-failed reads)
1379518 + 0 primary
0 + 0 secondary
13964 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
1268617 + 0 mapped (91.04% : N/A)
1254653 + 0 primary mapped (90.95% : N/A)
1379518 + 0 paired in sequencing
689759 + 0 read1
689759 + 0 read2
1241090 + 0 properly paired (89.97% : N/A)
1252774 + 0 with itself and mate mapped
1879 + 0 singletons (0.14% : N/A)
8814 + 0 with mate mapped to a different chr
7818 + 0 with mate mapped to a different chr (mapQ>=5)
1249362 + 0 in total (QC-passed reads + QC-failed reads)
1241090 + 0 primary
```

```

0 + 0 secondary
8272 + 0 supplementary
0 + 0 duplicates
0 + 0 primary duplicates
1249362 + 0 mapped (100.00% : N/A)
1241090 + 0 primary mapped (100.00% : N/A)
1241090 + 0 paired in sequencing
620545 + 0 read1
620545 + 0 read2
1241090 + 0 properly paired (100.00% : N/A)
1241090 + 0 with itself and mate mapped
0 + 0 singletons (0.00% : N/A)
0 + 0 with mate mapped to a different chr
0 + 0 with mate mapped to a different chr (mapQ>=5)

```

1.6 SNPs calling using data from 1613 and 1868 samples - bcftools mpileup ... | bcftools call ...

Print the help of bcftools mpileup

```
[ ]: #PUT COMMAND
```

```
[ ]: bcftools mpileup
```

Print the help of bcftools call

```
[ ]: #PUT COMMAND
```

```
[ ]: bcftools call
```

```
[ ]: #PUT COMMAND
```

```
[61]: bcftools mpileup --threads 8 -f ../REF/GCA_002220235.1_ASM222023v1_genomic.fna
      ↪*paired.bam | bcftools call --threads 8 -cv -Oz -o 1613_1868.bcf
```

Note: none of --samples-file, --ploidy or --ploidy-file given, assuming all sites are diploid

[mpileup] 2 samples in 2 input files

[mpileup] maximum number of reads per input file set to -d 250

```
[66]: ls -l
```

```

total 638540
-rw-r--r-- 1 jovyan users      521 Oct  3 14:06 1613.flagstat
-rw-r--r-- 1 jovyan users 161308863 Oct  3 14:07 1613.onlymappedpaired.bam
-rw-r--r-- 1 jovyan users   46104 Oct  3 14:09 1613.onlymappedpaired.bam.bai
-rw-r--r-- 1 jovyan users 174358429 Oct  3 14:03 1613.sorted.bam
-rw-r--r-- 1 jovyan users     519 Oct  3 14:32 1615.flagstat
-rw-r--r-- 1 jovyan users 143633305 Oct  3 14:30 1615.onlymappedpaired.bam

```

```
-rw-r--r-- 1 jovyan users      46184 Oct  3 14:30 1615.onlymappedpaired.bam.bai
-rw-r--r-- 1 jovyan users        512 Oct  3 14:32 1615.onlymappedpaired.flagstat
-rw-r--r-- 1 jovyan users 165320351 Oct  3 14:28 1615.sorted.bam
-rw-r--r-- 1 jovyan users   9123627 Oct  3 15:01 1616_1868.bcf
```

```
[68]: bcftools convert -Ov -o 1613_1868.vcf 1613_1868.bcf
```

```
[69]: ls -lh
```

```
total 660M
-rw-r--r-- 1 jovyan users   521 Oct  3 14:06 1613.flagstat
-rw-r--r-- 1 jovyan users 154M Oct  3 14:07 1613.onlymappedpaired.bam
-rw-r--r-- 1 jovyan users  46K Oct  3 14:09 1613.onlymappedpaired.bam.bai
-rw-r--r-- 1 jovyan users 167M Oct  3 14:03 1613.sorted.bam
-rw-r--r-- 1 jovyan users   519 Oct  3 14:32 1615.flagstat
-rw-r--r-- 1 jovyan users 137M Oct  3 14:30 1615.onlymappedpaired.bam
-rw-r--r-- 1 jovyan users  46K Oct  3 14:30 1615.onlymappedpaired.bam.bai
-rw-r--r-- 1 jovyan users   512 Oct  3 14:32 1615.onlymappedpaired.flagstat
-rw-r--r-- 1 jovyan users 158M Oct  3 14:28 1615.sorted.bam
-rw-r--r-- 1 jovyan users  8.8M Oct  3 15:01 1616_1868.bcf
-rw-r--r-- 1 jovyan users   36M Oct  3 15:34 1616_1868.vcf
```

1.7 Have a Look to the vcf created - head, tail

```
[70]: head *vcf
```

```
##fileformat=VCFv4.2
##FILTER=<ID=PASS,Description="All filters passed">
##bcftoolsVersion=1.13+htslib-1.13+ds
##bcftoolsCommand=mpileup --threads 8 -f
../REF/GCA_002220235.1_ASM222023v1_genomic.fna 1613.onlymappedpaired.bam
1615.onlymappedpaired.bam
##reference=file:///../REF/GCA_002220235.1_ASM222023v1_genomic.fna
##contig=<ID=F0082278.1,length=1352724>
##contig=<ID=F0082277.1,length=1122692>
##contig=<ID=F0082276.1,length=1091008>
##contig=<ID=F0082275.1,length=1037991>
##contig=<ID=F0082274.1,length=1019276>
```

```
[4]: tail 1613_1868.vcf
```

```
F0082259.2      71347      .      G      A      214.22      .      DP=144;VDB=0.492
924;SGB=29.6354;RPBZ=1.10961;BQBZ=1.78211;SCBZ=1.59266;FS=0;MQOF=0;AF1=0.5;AC1=2
;DP4=21,19,22,23;MQ=60;FQ=217.229;PV4=0.829011,1,1,1      GT:PL      0/0:0,120,255
1/1:255,135,0
F0082259.2      71427      .      T      G      999      .      DP=157;VDB=0.619
426;SGB=-1.38629;FS=0;MQOF=0;AF1=1;AC1=4;DP4=0,0,43,47;MQ=60;FQ=-160.489
GT:PL      1/1:255,129,0      1/1:255,141,0
```

```

F0082259.2      71435      .      G      T      214.22      .      DP=153;VDB=0.756
384;SGB=29.6354;RPBZ=1.95339;BQBZ=0.568112;SCBZ=0.887406;FS=0;MQOF=0;AF1=0.5;AC1
=2;DP4=24,20,19,26;MQ=60;FQ=217.229;PV4=0.291708,0.4528,1,0.14114      GT:PL
0/0:0,132,255      1/1:255,135,0
F0082259.2      71520      .      G      A      999      .      DP=128;VDB=0.651
316;SGB=-1.38598;FS=0;MQOF=0;AF1=1;AC1=4;DP4=0,0,39,34;MQ=60;FQ=-124.752
GT:PL      1/1:255,126,0      1/1:255,93,0
F0082259.2      71540      .      T      C      999      .      DP=119;VDB=0.403
379;SGB=24.0903;RPBZ=-1.18859;BQBZ=-0.483165;SCBZ=-
1.23039;FS=0;MQOF=0;AF1=0.5;AC1=2;DP4=16,15,17,20;MQ=60;FQ=217.229;PV4=0.807839,
0.307267,1,1      GT:PL      1/1:255,111,0      0/0:0,93,255
F0082259.2      71847      .      T      G      999      .      DP=73;VDB=0.9370
73;SGB=18.5451;RPBZ=-0.728589;BQBZ=-1.29364;SCBZ=-
2.47615;FS=0;MQOF=0;AF1=0.500004;AC1=2;DP4=4,13,19,10;MQ=60;FQ=217.229;PV4=0.013
3826,0.0457579,1,1      GT:PL      1/1:255,87,0      0/0:0,51,255
F0082259.2      71975      .      C      G      999      .      DP=79;VDB=0.3153
35;SGB=18.5451;RPBZ=-0.103735;BQBZ=-0.534962;SCBZ=-
0.971325;FS=0;MQOF=0;AF1=0.500032;AC1=2;DP4=4,10,10,19;MQ=60;FQ=217.23;PV4=1,0.2
49632,1,0.172793      GT:PL      1/1:255,87,0      0/0:0,42,255
F0082259.2      71996      .      T      G      999      .      DP=88;VDB=0.9650
53;SGB=-1.32836;FS=0;MQOF=0;AF1=1;AC1=4;DP4=0,0,18,32;MQ=60;FQ=-82.7539      GT:PL
1/1:255,99,0      1/1:255,51,0
F0082259.2      72324      .      C      T      214.22      .      DP=85;VDB=0.9444
27;SGB=15.7726;RPBZ=0.476356;BQBZ=1.90364;SCBZ=2.99153;FS=0;MQOF=0;AF1=0.5;AC1=2
;DP4=17,10,17,8;MQ=60;FQ=217.229;PV4=0.775612,1,1,0.49283      GT:PL
0/0:0,81,255      1/1:255,75,0
F0082259.2      72417      .      A      C      999      .      DP=90;VDB=0.6978
3;SGB=21.3177;RPBZ=0.133118;BQBZ=-2.21787;SCBZ=-
2.50838;FS=0;MQOF=0;AF1=0.5;AC1=2;DP4=12,9,18,15;MQ=60;FQ=217.229;PV4=1,0.010020
7,1,1      GT:PL      1/1:255,99,0      0/0:0,63,255

```

2 Download additional notebooks for exercices about mappin and SNP calling

```

[ ]: # Go into the correct directory to download notebook - cd command
cd /home/jovyan/NOTEBOOK/DAY3-4
wget https://itrop.ird.fr/trainings/notebook/DAY3-4/Day4_Mapping_Exercice1.
↪ipynb --no-check-certificate
wget https://itrop.ird.fr/trainings/notebook/DAY3-4/Day4_Mapping_Exercice2.
↪ipynb --no-check-certificate

```