**Practical exercises: File formats and QC**

An online version of this document can be found here <https://tinyurl.com/ybyq3rk3>. Please feel free to add comments if anything is unclear or incorrect. The answers to the exercises can be found at the end of this document. However, try to figure out answers yourself as that is the most effective way to learn.

**Exercise 1: SAM header line**

SAM/BAM format is the accepted standard format for storing NGS sequencing reads, base qualities, associated meta-data and alignments of the data to a reference genome. If no reference genome is available, the data can also be stored unaligned.

Download the SAM/BAM file specification document from<http://samtools.github.io/hts-specs>

([direct link](http://samtools.github.io/hts-specs/SAMv1.pdf)).

From reading page 4 of the SAM specification, look at the following line from the header of the BAM file:

@RG ID:ERR003612 PL:ILLUMINA LB:g1k-sc-NA20538-TOS-1 PI:2000 DS:SRP000540 SM:NA20538

**1.1** What does RG stand for?

**1.2** What is the sequencing platform/technology used to produced the reads?

**1.3** What is the lane ID?

(In sequencing terminology, a "lane" is the basic independent run of a high-throughput sequencing machine. Reads from one lane are identified by the same read group ID and the information about lanes can be found in the header in lines starting with @RG.)

**1.4** What is the expected fragment insert size?

**Exercise 2: SAM header and samtools**

Change directory to course\_data/Module2\_qc.

Samtools comprises a set of programs for interacting with SAM/BAM files. Type samtools with no parameters to display the list of available commands implemented in the program. Then type samtools view to display a detailed usage page.

Now use the samtools view command to print the header of the BAM file:

samtools view -H NA20538.bam | less -S

The -S option makes less display long lines without wrapping them. (You can toggle between the wrapping and non-wrapping mode by pressing -S at any time.)

**2.1** What version of the human assembly was used to perform the alignments? (Look for the genome assembly identifier AS.)

**2.2** How many lanes are in this BAM file? Remember that each lane is identified by a unique read group ID. Use the commands grep to parse the BAM header, looking for lines starting with @RG, and wc -l to count them.

(HINT: grep cannot operate in BAM itself. Use samtools view to allow grep to function.)

**2.3** What programs were used to create this BAM file? Look up the meaning of the @PG lines.

**2.4** What version of bwa was used to align the reads?

**Exercise 3: Alignment formats conversion**

You can use samtools to convert between SAM<->BAM and to extract regions of a BAM file. On the command line type

samtools view NA20538.bam | less -S

As explained above, the -S switch causes that long lines are truncated rather than wrapped, which makes the output more readable. Alternatively, the UNIX command cut can be used to extract only the columns of interest. (For example, the command cut -f1,4 prints only the first and the fourth columns of the input.)

**3.1** What is the name of the first read? Look up the QNAME field at page 5 of the SAM specification.

Note that although the specification makes a distinction between a "query template" (the physical sequenced molecule) and a "read" (the actual sequence obtained by the experiment), both are often used interchangeably.

**3.2** What chromosome and position does the alignment of the read start at?

**3.3** What is the mapping quality of the read?

We will convert a yeast BAM file to CRAM. In the data directory, there is a BAM file called yeast.bam that was created from *S. cerevisiae* Illumina sequencing data.

**3.4** Can you convert the BAM file to a CRAM file called yeast.cram using the samtools view command? First run the command without arguments to view the list of available options. For this exercise we will need -C, -T and -o. Note that the reference genome is stored in the file Saccharomyces\_cerevisiae.EF4.68.dna.toplevel.fa. Name the output file yeast.cram.

Since CRAM files use reference based compression, we expect the CRAM file to be smaller than the BAM file. What is the size of the CRAM file?

**3.5** Is your CRAM file smaller than the original BAM file?

**Exercise 4: VCF/BCF and bcftools**

VCF/BCF format is the accepted standard format for storing variant calls with supporting data. The official specification is available from<http://samtools.github.io/hts-specs>.

Bcftools comprises a set of programs for interacting with VCF/BCF files. You can use bcftools to convert between VCF<->BCF and to view or extract records from a region. Type bcftools without arguments to see the list of available commands. Then add name of any of the commands (for example, type bcftools view) to see the list of available options.

bcftools

bcftools view

Using the bcftools view command, print the header of the BCF file

bcftools view -h 1kg.bcf | less

and answer the following questions:

**4.1** What version of the human assembly the coordinates refer to?

**4.2** Can you convert the file called 1kg.bcf to a compressed VCF file called 1kg.vcf.gz using the bcftools view command? You will need the --output-file and --output-type options.

Similarly to BAM, the VCF/BCF format supports random access and can quickly retrieve records from any genomic region. For this, the file must be indexed.

**4.3** Index the BCF file. The index is stored in a newly created .csi file. (Note that the index file is never used directly, the program checks its existence and opens it automatically.) Then use the bcftools view command to extract records from a region by adding the option --regions 20:24042765-24043073.

Now we are able to extract complete records from any position or region. Can we extract individual fields as well? The versatile bcftools query command can be used to do that. Combined with standard UNIX commands, it gives a powerful tool for quick querying of VCFs. Try to answer the following questions with the help of the [manual page](http://samtools.github.io/bcftools/bcftools.html#query).

**4.4** How many samples are in the BCF? (Hint: check the -l option.)

**4.5** What is the genotype of the samples HG00107 and HG00108 at the position 20:24019472?

Use the bcftools query command with the following options:

--regions 20:24019472 to extract the VCF record at this position

--samples HG00107,HG00108 to extract the two samples

--format '%POS[ %GT]\n' to output the genotypes, printing first the position and then the genotypes separated by a space (the square brackets loop over samples)

**4.6** How many positions are there with more than 10 alternate alleles? First check the VCF specification and the VCF header (bcftools view -h). You will find that this information is encoded by the INFO/AC tag. Then extract all records with the INFO/AC value bigger than 10 using the --include 'AC>10' option and wc -l to count the lines.

**4.7** List positions where the sample HG00107 has a non-reference genotype and the read depth is bigger than 10. Similarly as above, use the bcftools query command with the following options:

--samples HG00107 to extract the sample

--include 'FORMAT/DP>10 & FORMAT/GT="alt"' to match positions with read depth bigger than 10 and with genotype containing an alternate allele. The ampersand symbol & requires that both conditions must be true in the same sample

--format '%POS[ %GT %DP]\n' to output position, the genotype, and the read depth.

Pipe the output into head to display only the first few lines.

**Exercise 5: Generate QC stats**

We will generate QC stats for two lanes of Illumina paired-end sequencing data from yeast. We will use the bwa mapper to align the data to the *Saccharomyces cerevisiae* genome (ftp://ftp.ensembl.org/pub/current\_fasta/saccharomyces\_cerevisiae/dna) and samtools stats to generate the stats.

**5.1** Read pairs are usually stored in two separate FASTQ files so that *n*-th read in the first file and the *n*-th read in the second file constitute a read pair. Can you devise a quick sanity check that reads in these two files really form pairs? The files must have the same number of lines and the naming of the reads usually suggests if they form a pair. The location of the files is

60A\_Sc\_DBVPG6044/lane1/s\_7\_1.fastq

60A\_Sc\_DBVPG6044/lane1/s\_7\_2.fastq

First check whether the read names are suggestive of a pair. For example, the reads in the first file can have the suffix /1 and the reads in the second file should have the suffix /2. Then check whether there is the same number of reads in both files.

Run the ./align.sh script to create the mappings. The script is very short, take a look inside using the command less ./align.sh. The script contains several commands, some are combined together using pipes. UNIX pipes is a very powerful and elegant concept which allows us to feed the output of one command into the next command and avoid writing intermediate files.

The script will produce the BAM file lane1.sorted.bam. Generate the stats including only primary alignments using the command

samtools stats -F SECONDARY lane1.sorted.bam > lane1.sorted.bam.bchk

Look at the output and answer the following questions:

**5.2** What is the total number of raw sequence reads?

**5.3** How many reads were mapped?

**5.4** How many read pairs were mapped to a different chromosome?

**5.5** What is the insert size mean and standard deviation?

Next we will create some QC plots from the output of the stats command using the command plot-bamstats which is of the samtools package:

plot-bamstats -p lane1-plots/ lane1.sorted.bam.bchk

In your web browser open the generated html file to view the graphs

firefox lane1-plots/\*.html

**5.6** How many reads have zero mapping quality (MQ)?

**5.7** Check the "Quality per cycle" graph. Which of the first fragments or second fragments are higher base quality on average?

**Answers to exercises:**

**1.1** **R**ead **G**roup

**1.2** Illumina, see the PL field

**1.3** ERR003612, see the ID field

**1.4** 2kbp, see the PI field

**2.1** NCBI build v37

**2.2** The command is

samtools view -H NA20538.bam | grep ^@RG | wc -l

**2.3** Use the command

samtools view -H NA20538.bam | grep ^@PG | less -S

Usually the alignments are processed with multiple programs. The @PG lines in this BAM file suggest that the reads were aligned using bwa, then GATK was used to recalibrate qualities and realign indels.

**2.4** Find VN field of the @PG line.

**3.1** The name of the first read is ERR003814.1408899, use for example a command like this

samtools view NA20538.bam | head -1 | cut -f1

**3.2** Chromosome 1, position 19999970

samtools view NA20538.bam | head -1 | cut -f3,4

**3.3** Q23

samtools view NA20538.bam | head -1 | cut -f5

**3.4** Use the command

samtools view -C -T Saccharomyces\_cerevisiae.EF4.68.dna.toplevel.fa

-o yeast.cram yeast.bam

**3.5** Use the command

ls -lh yeast.bam yeast.cram

**4.1** Lookup the ##reference line

**4.2** Use the command

bcftools view -O z -o 1kg.vcf.gz 1kg.bcf

**4.3** Use the following commands:

bcftools index 1kg.bcf

bcftools view -H -r 20:24042765-24043073 1kg.bcf | less -S

**4.4** Use bcftools query -l 1kg.bcf to get list of samples and wc -l to count the lines

bcftools query -l 1kg.bcf | wc -l

**4.5** The complete command is

bcftools query -r 20:24019472 -s HG00107,HG00108 -f '%POS [ %GT]\n' 1kg.bcf

**4.6** The command can look like this:

bcftools query -i 'AC>10' -f '%POS\n' 1kg.bcf | wc -l

**4.7** The complete command is

bcftools query -s HG00107 -i 'FORMAT/DP>10 & FORMAT/GT="alt"' -f '%POS [ %GT %DP]\n' 1kg.bcf | head

**5.1** You can use the following commands

head 60A\_Sc\_DBVPG6044/lane1/s\_7\_1.fastq | grep ^@

head 60A\_Sc\_DBVPG6044/lane1/s\_7\_2.fastq | grep ^@

wc -l 60A\_Sc\_DBVPG6044/lane1/\*.fastq

**5.2** Look inside the file and locate the field "raw total sequences". To extract the information quickly from multiple files, commands similar to the following can be used:

grep ^SN lane\*.sorted.bam.bchk | awk -F'\t' '$2=="raw total sequences:"'

**5.3** Locate the field "reads mapped" or use the command

grep ^SN lane\*.sorted.bam.bchk | awk -F'\t' '$2=="reads mapped:"'

**5.4** Locate the field "pairs on different chromosomes" or use the command

grep ^SN lane\*.sorted.bam.bchk | awk -F'\t' '$2=="pairs on different chromosomes:"'

**5.5** Locate the "insert size" fields