**PROGRAM: MSC BIOINFORMATICS, MAKERERE UNIVERSITY**

**STUDENT: KATONGOLE MIKE**

**REG NO: 2022/HD07/5010U**

**INSTRUCTOR: IBRA LUJUMBA**

**COURSE: BIOUNIX AND SHELL SCRIPTING**

**ASSIGNMENT ANSWERS AND FILE EXPLANATIONS**

* + - 1. Variant Call Format (VCF) file is a text file that contains information about genetic variations in a genome population. It is a standard file format for storing genetic variations data. The data stored in the in the VCF file includes information about the position of a variant on the genome, the reference and alternate alleles, and various quality metrics, such as the read depth and quality scores.
      2. The header section of the VCF file contains meta information about the file, such as the file format version, the sample information, and the reference genome used. It is preceded by a line starting with ‘##’ and contains one or more lines of information.
      3. The command **bcftools query -l file.vcf | wc -l** will find out how many samples are in the file.

They were 6 in this file

* + - 1. The number of variants in the file can be determined by counting the number of lines in the file that do not start ‘##’ the command **bcftools query -f ‘ALT\n’ file.vcf | wc -l** or **bcftools view -H file.vcf | wc -l** are used.

They were 398246 variants in the file

* + - 1. To extract the chromosome, position QualBY depth, RMSMappingQuality fields, command **bcftools query -f '%CHROM\t%POS\t%INFO/QUALbyDEPTH\t%INFO/RMSMappingQuality\n' file.vcf > tabdel.vcf**
      2. To extract data that belongs to chromosome 2,4 and MT, you can uses command line tools like awk and grep, Case: awk command was used **awk '$1=="2" || $1=="4" || $1=="MT"' sample.vcf** OR command **bcftools view -r 2,4,MT file.vcf > output.vcf**
      3. To print out variants that do not belong to chr20:1-30000000, you can use command line tools like awk, grep and sed. Or **bcftools view -R -t ^20:1-30000000 file.vcf**
      4. To extract variants that belong to SRR13107019, you can use command line tools awk and grep OR bcftools e.g

**bcftools view -s SRR13107019 file.vcf > srr9variants.vcf**

or

**bcftools query -f '%CHROM\t%POS\t%REF\t%ALT\n' -s SRR13107019 sample.vcf > srr9variants.vcf**

* + - 1. To filter out variants with a QualByDepth above 7, you can use command line tools like awk and grep also bcftools can be used e.g.

**bcftools filter -i 'INFO/QUALbyDEPTH>7' file.vcf > qbdvariants.vcf**

or

**awk -F '\t' '{if ($6>=7) print $0}' sample.vcf > qbdvariants.vcf**

* + - 1. The number of contigs referred to in the file can be determined by looking at the header section of the file and counting the number of contigs listed in the “contig” column

The following Commands can be used,

**bcftools view -h file.vcf | grep -o -w 'contig=[^;]\*' | sort | uniq | wc -l**

or

**grep -c "^##contig" sample.vcf**

The number of contigs in this case is **2211**

* + - 1. The eighth and ninth columns of the file contain information about the genotype of the sample for a given variant. The column contains the genotype format and the ninth column contains the genotype data for each sample in the file.
      2. To extract data on the read depth of called variants of sample SRR13107018, you can use command line tools like awk and grep or bcftools e.g

**bcftools query -f '%DP\n' -s SRR13107018 sample.vcf > srr8.vcf**

* + - 1. To extract data on the allele frequency of alternate allele frequency of alternate allele.

**bcftools query -f '%CHROM\t%POS\t%AF\n' sample.vcf > alleles.vcf**

**SAM FILE.**

1. The format of a SAM file is a tab-delimited text, with each line representing a read alignment and containing a number of fields, such as the read name, the reference name, the alignment start position, and the alignment quality. The file also contains a header section that specifies metadata about the file, such as the reference genome used for alignment.
2. The headers section of a SAM file contains information about the reference genome used for alignment, the program that was used to generate the alignments, and the options used. It starts with the “@” symbol and contains lines starting with “@HD”, “@SQ”, “@RG”, “@PG” and “@CO”
3. The number of samples in the file will depend on the sequencing experiment and the number of samples that were processed together. The information about the sample can be found in the “@RG” tag of the header section.

The Commands can be;

**grep -E '^@RG' sample.sam | awk '{print $2}' | awk -F ':' '{print $2}' | sort | uniq | wc -l**

**or**

**grep '^@RG' file.sam | cut -f2**

The number of samples in the file are **249.**

1. To know the number of alignments in the file, we can still use samtools which will print the number of non-header lines in the file, which represents the alignments.

The commands can be;

**samtools view -F 4 sample.sam | wc -l**

Or

**grep -v '^@' file.sam | wc -l**

The number of alignments were **35511**

You can use the samtools command line tool to obtain summary statistics for the alignments in the file. For example, we can use the **samtools flagstat file.sam** command to get a summary of the flags for each alignment, such as the number of reads that are mapped, unmapped, or paired-end.

1. You can use the command **samtools flagstat file.sam** to get summary statistics for the alignments in the file.
2. You can use the command **head -n1 file.sam | tr '\t' '\n' | wc -l** or **awk '{print NF}' sample.sam | sort -nu | wc -l** to count the number of fields in the file.

The number of fields in this file are **9**

1. To print all lines in the file that have @SQ and sequence name tag beginning with NT\_, you can use the command **grep '^@SQ.\*NT\_' file.sam**
2. To print all lines in the file that have @RG and LB tag beginning with Solexa, you can use the command **grep '^@RG.\*LB:Solexa' file.sam**
3. To extract primarily aligned sequences and save them in another file, you can use the command **samtools view -F 4 file.sam > output.sam**
4. To extract alignments that map to chromosomes 1 and 3, you can use the command **samtools view -b file.sam 1 3 > output.bam**

Or

**awk '$1 !~ /^@/ && ($3 == "1" || $3 == "3")' sample.sam | samtools view -Sb - > alignments\_chr1\_3.bam**

The samtools view command filters the alignments based on the flags, -f 3 (only primary alignments), -F 4(exclude unmapped), -F 256 (exclude supplementary alignments) and -F 2048 (exclude secondary alignments) and then it saves the output to a BAM file using the samtools sort command.

1. To obtain unmapped reads from the file, you can use the samtools view command with the flag -f 4

**samtools view -f 4 sample.sam > unmapped\_reads.sam**

1. You can use the command grep -c to count the number of reads that are aligned to the chromosome 4

Command; **grep -c "^4\t" sample.sam**

1. The second column of the SAM is the flag field, which contains information about the read alignments such as whether it is paired-end, whether it is the first of second in pair, and whether it is mapped or unmapped. The sixth column is the mapping quality of the reads alignment.
2. To extract all optional fields of the file and save them in “optional\_feilds.txt”, you can use the awk command.

**awk '{for(i=11;i<=NF;i++) print $i}' sample.sam > optional\_fields.txt**