**MAKERERE UNIVERSITY**

**COLLEGE OF HEALTH SCIENCES**



**MSc BIOINFORMATICS**

**COURSE: BASH AND SHELL**

|  |  |
| --- | --- |
| **OKELLO ISAAC OPIO** | **2022/ HD07/2052U** |

**TASK:**

You are provided with two files (A VCF and a SAM file). The files can be downloaded from here

[(https://drive.google.com/drive/folders/11UD52i99CaCSBEJFNb8Y1afo9p3hL8cL?usp=shari ng](https://drive.google.com/drive/folders/11UD52i99CaCSBEJFNb8Y1afo9p3hL8cL?usp=sharing)). Use BCFtools, SAMtools and bash utilities to answer the questions that follow.

Make your submissions on your GitHub and send the link to the repo via email to ibra.lujumba@gmail.com. Your submissions should include commands/scripts used to obtain answers to the questions as well as answers to the questions.

**Deadline:** 26th January 2023

**Manipulating VCF files**

1. Describe the format of the file and the data stored

Variant Call Format (VCF) is a format for storing variations between a reference genome and sequences aligned to it, based on SAM/BAM alignments. This type of file is used to store genetic variation data, such as single nucleotide polymorphisms (SNPs), insertions and deletions (indels), and structural variants. It has a text format, and it starts with a header section that contains meta-information about the file and the data it contains. VCF files begin with a header section: lines in the header section begin with “##”. The last line in the header section begins with #; this line gives the headers of the columns used in the VCF file.

1. What does the header section of the file contain?

The header section of a VCF file provides a summary of the format and origin of the data contained in the file. This section starts with lines that begin with "##" and contains key-value pairs that specify information such as the version of the VCF format used, the reference genome that the variants were called against, and details about the software and pipeline that were used to generate the data.

1. How many samples are in the file

The code used was: bcftools query -l sample.vcf.gz | wc -l

The answers were 6.

1. How many variants are in the file

I found 398246 variants when I used the code: bcftools query -f '%ALT\n' sample.vcf | wc -l

1. How would you extract the chromosome, position, QualByDepth and RMSMappingQuality fields? Save the output to a tab-delimited file

You can use the code: bcftools query -f '%CHROM\t%POS\t%INFO/QD\t%INFO/MQ\n' sample.vcf > positions.txt

The output is in the file named “position.txt”.

1. Extract data that belongs to chromosomes 2,4 and MT

Using the code: awk '$1 ~ /^(2|4|MT)$/ {print $0}' sample.vcf > chrom\_data.vcf

The output was in the output file chrom\_data.vcf

1. Print out variants that do not belong to chr20:1-30000000

The output was stored in file named “nonChr-var.vcf” and this code was run:

awk '$1 != "20" || ($1 == "chr20" && ($2 < 1 || $2 > 30000000)) \

{> {print $1, $2, $4, $5}' sample.vcf > nonChr-var.vcf

1. Extract variants that belong to SRR13107019

To get variants, use the code: bcftools query -f '%CHROM\t%POS\t%REF\t%ALT\n' -s SRR13107019 sample.vcf > extracted\_variant.txt

The output is named “extracted\_variant.txt”

1. Filter out variants with a QualByDepth above 7

You can use the code: vcftools --vcf sample.vcf --minDP 7 --recode --out Qual\_output

And the output file was specified to be “Qual\_output”

The output also stated that after filtering, kept 398246 out of a possible 398246 Sites.

1. How many contigs are referred to in the file. Check the header section

There are 2211 contigs.

Use the code grep -c "^##contig" sample.vcf

1. Comment on the eighth and ninth columns of the file

The eighth column of a VCF file typically contains information about the genotype of the sample being called, while the ninth column and following columns contain the sample-specific information, including the genotype likelihoods, the phred-scaled quality of the genotype call, the depth of coverage and other annotation tags.

1. Extract data on the read depth of called variants for sample SRR13107018

You can use the code: bcftools query -f '%DP\n' -s SRR13107018 sample.vcf > depth\_SRR18

1. Extract data on the allele frequency of alternate alleles. Combine this data with the chromosome and position of the alternate allele

To get and output in a file named “allele\_data.txt”, use the code: bcftools query -f '%CHROM\t%POS\t%AF\n' sample.vcf > allele\_data.txt

**Manipulating SAM files**

1. Describe the format of the file and the data stored

The sequence Alignment/Map (SAM) is a text-based alignment format that supports single- and paired-end reads produced by different sequencing platforms. The SAM format consists of a header and an alignment section. Headings begin with the @ symbol, which distinguishes them from the alignment section. All lines in a SAM file are tab-delimited. Alignment sections contain 11 mandatory fields, with other fields being optional. Although the mandatory fields must be present, their value can be a \* or an 0 depending on the field.

1. What does the header section of the file contain?

It has a header that begin with the @ symbol, which distinguishes them from the alignment section. This section contains information about the entire file and additional information for alignments.

1. How many samples are in the file

It is 249 samples, when this code is used: grep -E '^@RG' sample.sam | awk '{print $2}' | awk -F ':' '{print $2}' | sort | uniq | wc -l

1. How many alignments are in the file

Using code: samtools view -F 4 sample.sam | wc –l

I got 35511 alignments.

1. Get summary statistics for the alignments in the file

I used the code: samtools flagstat sample.sam > summary\_sam.txt

And had an output in the file named “summary\_sam.txt”

1. Count the number of fields in the file

Use the code: awk '{print NF}' sample.sam

1. Print all lines in the file that have @SQ and sequence name tag beginning with NT\_

I used the code: grep '@SQ.\*NT\_' sample.sam

1. Print all lines in the file that have @RG and LB tag beginning with Solexa

Use the code: grep "@RG.\*LB:Solexa" sample.sam

1. Extract primarily aligned sequences and save them in another file

Use the code: awk '$1 !~ /^@/ && $2 == "99" || $2 == "83"' sample.sam > aligned\_seq.sam

The output is named “aligned\_seq.sam”.

1. Extract alignments that map to chromosomes 1 and 3. Save the output in BAM format

I used the code to attempt to produce the output as “map\_chr1.3.bam”: awk '$1 !~ /^@/ && ($3 == "1" || $3 == "3")' sample.sam | samtools view -Sb - > map\_chr1.3.bam

However, the second part of the code which performs the conversion of SAM to BAM seems not to be working.

1. How would you obtain unmapped reads from the file

Use the code: samtools view -f 4 sample.sam > reads\_unmapped.sam

This creates and output with a file name “reads\_unmapped.sam”

1. How many reads are aligned to chromosome 4

The code: grep -c "^4\t" sample.sam

Gave an output of 0 reads

1. Comment of the second and sixth column of the file

The second column of the SAM contains the name of the reference sequence that the reads align to. Then sixth column describes the CIGA string which describes the alignment of the read to the reference sequence.

1. Extract all optional fields of the file and save them in “*optional\_fields.txt*”

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