**BIOMI 609 Computational Genomics and Bioinformatics**

**Spring 2022**

**San Diego State University**

**Dr. Arun Sethuraman**

**Lab 2 - Alignment to a reference genome, variant calling**

Now that we have learned the theory behind variant calling and alignment to a reference genome, we will spend today’s lab playing with some memory efficient tools for this purpose. As before, we will work with SARS-CoV2 genomes (downloaded as a FASTQ from Lab 1), and the SARS-CoV2 reference genome (Wuhan-1).

**Exercise 0 - Installing pre-requisite tools: bwa-mem2, pyfaidx**

Login to your VM via Jetstream.

First, will install bwa-mem2 (<https://github.com/bwa-mem2/bwa-mem2>), a tool from Heng Li’s group that is the most memory efficient BWT tool out there. To use the precompiled libraries, move into your Tools folder created last week, then:

curl -L <https://github.com/bwa-mem2/bwa-mem2/releases/download/v2.0pre2/bwa-mem2-2.0pre2_x64-linux.tar.bz2> | tar jxf -

This should generate bwa-mem2 as a precompiled library. Now go ahead and set the PATH as before to make sure that bwa-mem2 is accessible.

pip install pyfaidx --user

Set PATH to pyfaidx

**Exercise 1 - Downloading SARS-CoV2 sequences from GISAID**

At this point, everyone should have created an account on GISAID - so open a Firefox window inside your VM, login to their website, click on “Search”. This should take you to a query form, where you can search for genomes by various attributes. Get creative here - search by lineage (e.g. alpha, beta, delta, omicron, etc)., or search by country, dates, etc. Make sure to obtain only “complete” and “high coverage” genomes. Thereon, select a handful of those sequences, and click on “Download”. Here, make sure to download it as a FASTA file. This should produce a FASTA file containing all the selected sequences and download it to your VM (usually these should be in the Downloads folder). In my case, I obtained several delta sequences, and just renamed the FASTA as deltas.fasta (using the mv command in Unix).

**Exercise 2 - Creating a BWT from your reference genome, aligning it to your FASTA**

Thereon, move into the folder with your SARS-CoV2 FASTA file(s) and the reference genome. To generate a BWT from the reference genome:

bwa-mem2 index reference.fasta

mkdir deltas

mv deltas.fasta deltas/

cd deltas

#this is to align them for phylogenetic construction, which we’ll do later

clustalw2 -INPUTFILE=deltas.fasta -OUTPUTFILE=deltas\_aligned.fasta

#this is to split the deltas.fasta file into separate FASTA files to be aligned to the reference

faidx --split-files deltas.fasta

Now since we have to individually align these FASTA’s to the reference, we are going to write a loop. Let’s actually do this in a SLURM Unix shell script and run it instead of running things on the command line!

Note that you can use any text editor of your choice - I like VIM, but you’re welcome to use any other editor.

Create a new file, and edit it with the header followed by the loop:

#!/bin/bash

for FILE in \*

do

bwa-mem2 mem ../reference.fasta $FILE > $FILE.sam

samtools view -S -b $FILE.sam > $FILE.bam

#samtools view out.bam | head

samtools sort $FILE.bam -o $FILE.sorted.bam

samtools index $FILE.sorted.bam

done

Now save this file as runsamtools.sh. Thereon, we have to set permissions for this file:

chmod +x runsamtools.sh

You also need to write a samples.ploidy file with \* \* \* \* 1 in n lines, where n = # of samples - this is important, since samtools can incorrectly guess the ploidy (here 1).

To do this, you can use a simple hack:

#to find the number of FASTA files

ls \*.fasta | wc -l

This should print the number of FASTA files, let’s say this is n, then you just write a simple loop:

for i in {1..n} do

echo “\* \* \* \* 1” >> samples.ploidy

done

Then you can run this script as:

./runsamtools.sh

From there, we will use the following commands to call variants:

bcftools mpileup -f ../reference.fasta \*.sorted.bam | bcftools call -mv -Ov --ploidy-file samples.ploidy -o delta.bcf

bcftools view delta.bcf > delta.vcf

Voila! Observe the VCF file and summarize it.