Mapping to a reference genome

In this document, normal text is for reading, this font is for typing. You should be able to copy and paste the commands into your terminal window.

**Setup**

Install tools: Sickle, Fastqc, samtools.

We are going to set up self-contained ‘environment’ for this workshop to make things simple. To do this we will install a program called ‘conda’ which already comes with many of the tools we need (much quicker/easier than installing all tools manually).

Install conda:

Details here (for Mac users): <https://conda.io/docs/user-guide/install/macos.html>

Make sure conda know where to find the ‘activate’ command

*# Edit this path for your laptop (may be as simple as swapping ‘seanmeaden’ for your username*

source /Users/seanmeaden/miniconda3/bin/activate

Make a new conda environment:

*# Install some conda channels*

*# A channel is where conda looks for packages*

conda config --add channels defaults

conda config --add channels conda-forge

conda config --add channels bioconda

*# Create a new conda environment called ngs*

conda create -n ngs python=3

*# activate the environment*

conda activate ngs

Run conda and install all the tools we will need shortly:

conda activate ngs

conda install samtools

conda install bedtools

conda install bwa

conda install picard

conda install fastqc

conda install bcftools

**Part 1. Get data and quality filter**

cd ~/Desktop

mkdir mapping\_tutorial/

cd mapping\_tutorial/

wget <https://www.dropbox.com/s/1hmjqusgx2fsreh/H1_sub_R1.fastq.gz?dl=0>

wget <https://www.dropbox.com/s/r1fswkypch9xm9c/H1_sub_R2.fastq.gz?dl=0>

wget <https://www.dropbox.com/s/xmnz82868p8zrtn/DMS3_ncbi.fasta?dl=0>

# Need to rename them (weird suffix added by dropbox)

mv H1\_sub\_R1.fastq.gz\?dl\=0 H1\_sub\_R1.fastq.gz

mv H1\_sub\_R2.fastq.gz\?dl\=0 H1\_sub\_R2.fastq.gz

mv DMS3\_ncbi.fasta\?dl\=0 DMS3\_ncbi.fasta

# Unzip files (\* is a wildcard i.e. ‘unzip all files that end with .gz)

gunzip \*.gz

# Check quality of sequence data using FastQC

# Make directory for FastQC output files:

mkdir fastqc\_output

# Run for one of the sequence files:

fastqc -o fastqc\_output H1\_sub\_R1.fastq

# Check output:

open fastqc\_output/H1\_sub\_R1\_fastqc.html

Seeing as the quality is fine we are going to skip quality filtering. In real-life we will rarely do this. I like a tool called ‘Sickle’ which is available here:

https://github.com/najoshi/sickle

**Part 2. Map reads to reference genome and call SNPs**

Let’s map the reads to the reference genome using a program called BWA (Burrows-Wheeler alignment). An alternative is called Bowtie2- I wouldn’t use anything else. We first need to ‘index’ the reference genome (this allows ‘random access’ so the program doesn’t need to read the file from top to bottom, thus speeding the process immensely). BWA takes an array of input file formats (Sanger, PacBio, MinIon) but we will usually have paired-end Illumina reads. The aligner makes use of the paired reads to make better alignments than if you mapped them individually.

bwa index DMS3\_ncbi.fasta

# Run alignment with paired end reads – note positional arguments.

bwa mem -M -t 1 DMS3\_ncbi.fasta H1\_sub\_R1.fastq H1\_sub\_R2.fastq > aln.sam

The output is a SAM file. We now go through some conversion steps to get a usable dataset.

samtools view -bT DMS3\_ncbi.fasta aln.sam > aln.uns.bam

samtools sort aln.uns.bam aln.sorted

# Remove intermediate files

rm aln.uns.bam

rm aln.sam

# Again, we need to index the file

samtools index aln.sorted.bam

Next we will call SNPs / Indels based on the alignment file we have.

Note that this is actually 2 commands. We ‘pipe’ the output of the first command (samtools mpileup) into the second (bcftools view). Investigate the flags used (-u etc) as they dictate the different options. Also investigate other SNP calling methods. This is a simple but possibly outdated method (could use FreeBayes, GATK HaplotypeCaller or others).

samtools mpileup -u -g -f DMS3\_ncbi.fasta aln.sorted.bam | bcftools view -vcg - > file\_raw.vcf

**Part 3. Sanity check- do SNPs look real?**

Let’s pick a SNP and see how the reads stack up at that position using Tablet.

**Conclusion**

Hopefully this gives an overview of a SNP calling pipeline. Almost every command has various options and there are numerous strategies to ID mutants- and none are perfect. By wrapping all the above commands into a single BASH script we could automate the process. This is useful for repeating with different parameters, or if we have many samples.