Genome 373
Metagenomic alignment workshop
Day 1: May 31, 2017

Part 1: Getting started with the Docker container

1) Install Docker: https://www.docker.com/community-edition#/download

2) Open a terminal window and make a directory from which you'll run the container. You'll need it to be able to access output data after you close the container.

mkdir [new_directory_name]

3) With the Docker app running, navigate to that directory in the terminal and pull an image:

cd [new_directory_name]
docker pull cnoecker/workshop373:final

This will download an image from the online Docker hub. You will need a few (~5-8) GB of storage free on your computer.

4) Start an interactive container with the image by typing the following command in your terminal:

docker run -it -v $PWD:/data/output cnoecker/workshop373:final

5) The container includes an installation of the following programs:

- fastq-dump, a utility for downloading sequence data from the NCBI Short Read Archive.
- fastqc, which is a short read quality analysis tool. Fastqc takes FASTQ files as input and outputs an HTML file summarizing read quality.
- bwa, which is a short read mapper based on the Burrows-Wheeler transform. Bwa takes a reference sequence and reads in FASTQ format, producing a Sequence Alignment Map (sam) file.
- samtools, which is a suite of software for manipulating sam files.
- bedtools, which is a suite of software for analyzing genomic data.

It also includes data:

- In the “data/refs” directory: a reference file containing 3 pathogen genomes, and various associated files

Take a minute to explore the directory structure inside the container.

6) To close the container, you just type "exit". You will only be able to access files you created within the container if they were placed in the "data/output" directory, which was linked with the directory you started the container in. You may also want to run "docker system prune" when you're done to clean up extraneous files and processes from the container run.

Part 2: Download samples, quality-check and align reads

Each group will analyze 2 metagenomic samples from Joensen et al 2017. Work through Parts 2-4 with a single sample first, and if you have time, go back and repeat your analysis on the second sample.

Group 1: ERR1543975, ERR1543950
Group 2: ERR1543953, ERR1543982
Group 3: ERR1543986, ERR1543952
Group 4: ERR1543973, ERR1543944

1) Download your data files and make them accessible later by typing the lines of code below in your terminal.

cd output
fastq-dump --split-spot --gzip [sampleID]

While the files are downloading (some of them are larger than others), you can do the following:

- Read the abstract of the study (linked above) that generated this dataset. What kind of samples are these? What was the research question?
- Look over this link on defining variables in the terminal using the Bash language. You'll find it useful for understanding the code you'll be running for the rest of the workshop.

2) The reference genomes are located in the "data/refs" directory. Take a look at the reference genomes. How are they formatted? Notice that we have already used BWA to generate an index for these reference genomes.

3) Define variables to represent file names.
We'll define variables in the bash shell (command line) to make the code we provide general for any of the samples provided to the groups. First, define bash shell variables for the files we'll be working with:

sampleID="ERRxxxxxxx" #Your first sample ID
samplefile="$sampleID".fastq.gz
samfile="$sampleID".sam
bamfile="$sampleID".bam

4) Check the quality of your samples by running a FastQC analysis with the following command:

cd ..
fastqc --outdir=output/ output/"$samplefile"

We'll return to the output file produced by the command above.

5) Next, use bwa to align the reads in the FASTQ file to the reference, generating a sam file. We have provided most of the line of code you will need to map the reads below. However, bwa has three different modes designed for different types of reads. Read the bwa FAQ on Sourceforge and decide which mode is appropriate. Then replace COMMAND below with the

appropriate option (hint: typing bwa without any additional options will give you a list of all possible commands).

bwa COMMAND refs/pathogen_ref.fna output/$samplefile > $samfile

6) While bwa is aligning the reads, open the html file produced by FastQC. Make sure you understand what each plot is showing.

- At the FastQC website, there are example logs from "good Illumina data" and "bad Illumina data". How do your samples compare?
- Looking at the per-base quality scores, what pattern do you notice? What do you think might explain the pattern you see?

7) Once bwa has finished aligning the reads, have a look at the sam file using head or more. Each line shows where a read maps, and gives a variety of additional useful information. What does each column represent? You can find the full specification here.

8) For downstream analysis, the human-readable sam format is inefficient. So, use samtools to generate a bam file, which is simply a binary version of the sam file:

samtools view -bS $samfile > $bamfile

The samtools "view" command can be used to get information on how our reads mapped. For example, using -c flag will print the number of matching records. You can count the number of mapped reads using the following command:

samtools view -c -F 4 $bamfile

Alternately, you can determine the number of unmapped reads:

samtools view -c -f 4 $bamfile

Determine the ratio of mapped to unmapped reads. Does this ratio surprise you? Why or why not?

Next, use samtools to produce a bam file with just the mapped reads, since those are the ones of interest:

samtools view -bhF 4 $bamfile > "$sampleID"_mapped.bam

For downstream analyses, it's useful to sort and index the mapped reads. Sorting arranges the reads in ascending order based on their left-most coordinates (e.g. where the left end of the read maps to the reference). Indexing enables fast searching of the sorted, mapped reads and is required for visualization of the mapped reads using IGV (see below).

samtools sort "$sampleID"_mapped.bam "$sampleID"_mapped_sorted
samtools index "$sampleID"_mapped_sorted.bam

Part 3: Visualize the alignment with IGV

- Download and launch the Java app at this link: http://software.broadinstitute.org/software/igv/download
- Copy each required file below to the "output" directory in your docker container. Only files in this directory are visible outside the container.
- Load in the reference file (pathogen_ref.fna) using the Genomes -> Load genome from file… dropdown
- Load in the gene file (pathogen_ref.bed) and your sorted and indexed .bam file (note, you will also have to copy the index .bai file to the output directory) using the File -> Load from file… dropdown.

First, get familiar with IGV. You may want to check out the IGV tutorial. In particular, the page on visualizing sequence tracks may be helpful.

- You should see three tracks in IGV. What does each one show? Note you will need to zoom in for IGV to display two of the tracks.
- Try zooming in so you can see a region of ~100 nucleotides. You should be able to see each base of the reference, as well as the read pileup from sequencing.
- Make sure you know how to access each of the 3 reference genomes. Are there any visibly apparent differences in how many reads are shown to align to each one?
- Find a few single nucleotide variants and see if you can figure out whether they are real or sequencing artifacts by examining the quality of each read (hover the mouse over the appropriate position in the read to see additional information, including the Phred score (QV)).
- These are paired-end reads, but the bwa alignment we ran just aligned each element of each pair independently. Does it look like most pairs map to the same genomic location?

Part 4: Determine which species predominate in each sample

For each sample, your goal is to count the number of reads mapping to each of the three reference genomes. This can be accomplished in a variety of ways. Split your team into two groups and:

- Use samtools to figure out how many reads map to each genome. As a first step, have a look at each of the samtools commands (just type samtools on the command line) and think about which one might be helpful. You can find out more about a particular command by typing: "samtools [command] -h". Then, figure out how to use it to accomplish your goal.
- Write a Python script to figure out how many reads map to each genome. Note that you will need a sam (not bam) file with mapped reads. So, your first step will be to use samtools to convert your mapped/sorted bam file into sam format. Then, figure out which column of the sam file has the information you need and write a Python script to go through the sam file line by line.

In both cases, you will need to know the string identifying each of the three genomes. This string can be found in the header of the sam file itself, and also in the information in the "refs/" directory.

Do the results from each method match? Make a plot (e.g. using Excel) showing the number of reads mapping to each species for both samples. What conclusions can you draw? Is it fair to compare the number of reads across species? Why or why not?