Exercise 2: Troubleshoot Emu output - Extract selected reads etc

This exercise requires that you have first ran Exercise 1, as it will be using data resulting from running Emu.

In this exercise we will be extracting reads for a particular species, as a way to demonstrate how you can investigate specific classifications manually when needed.

1. Enter the folder where you ran Exercise 1

   E.g:

   ```
   cd ex1
   ```

2. Activate the Pixi environment you created there:

   ```
   pixi shell
   ```

3. Install samtools and seqkit via Pixi, as we will need them later:

   ```
   pixi add samtools seqkit
   ```

4. Locate the `.sam`-file in the emu output:

   ```
   find -name '*.sam'
   ```

5. Now you can have a look at this file with samtools:

   ```
   samtools view ./emuout/sample01.fq_emu_alignments.sam | less -S
   ```

   You will see that the first rows, and the first three columns look like this:

   ```
   f56c2a13-e09f-424e-9151-b548e0c36ff4    0       1639:emu_db:33855
   f56c2a13-e09f-424e-9151-b548e0c36ff4    256     1639:emu_db:33891
   f56c2a13-e09f-424e-9151-b548e0c36ff4    256     1639:emu_db:33875
   f56c2a13-e09f-424e-9151-b548e0c36ff4    272     1639:emu_db:33896
   f56c2a13-e09f-424e-9151-b548e0c36ff4    272     1639:emu_db:33952
   ```

The third column here contains an NCBI tax ID as its first field, which means we can extract reads based on this.

6. We can start by checking which are the most common taxids, by extracting this field, sorting, counting and sorting again based on occurence:

```
samtools view emuout/sample01.fq_emu_alignments.sam \
    | cut -f3 \
    | cut -d: -f1 \
    | sort \
    | uniq -c \
    | sort -nr \
    | less -S
```

Then we might see results like:

```
7040 1639
6695 1280
6001 1613
5784 1423
5690 1351
2333 28901
2182 562
1310 287
1168 1680
 608 1127744
```

Here, we see the counts on field 1, and the tax ids in field 2.

7. Now, the reason we would want to extract reads from some specific species would most reasonably be because we have seen that they have lower alignment quality of some kind, something we can not easily see from the Emu output alone, but which can be easier seen by the metrics calculated by the TranaVy reporting tool we will use in Exercise 3.

   But so for the time being, we just imagine that we want to extract those “Salmonella enterica” reads with ID 28901 (You can use the NCBI Taxonomy lookup page to check that).

   To extract those, we can use a bit of awk-fu, combined with filtering for non-zero mapping qualities, to only get those reads that got a unique alignment:

```
samtools view -e "mapq > 0" emuout/sample01.fq_emu_alignments.sam \
  | awk -F"\t" '$3 ~ "28901:emu_db.*"' \
  | less -S
```

   We see that we don’t get so many reads now, which is nice if we want to try to blast them.

   What we need to do to turn this into a fastq file, is to extract three things:

   - The read ID

- the sequence
- and the quality field.

This we can do by extending the command above with a `cut` command for fields 1,10 and 11:

```
samtools view -e "mapq > 0" emuout/sample01.fq_emu_alignments.sam \
  | awk -F"\t" '$3 ~ "28901:emu_db.*"' \
  | cut -f 1,10 \
  | less -S
```

This can finally be piped to `seqkit tab2fx` to convert it to fasta:

```
samtools view -e "mapq > 0" emuout/sample01.fq_emu_alignments.sam \
  | awk -F"\t" '$3 ~ "28901:emu_db.*"' \
  | cut -f 1,10 \
  | seqkit tab2fx > salmonella_enterica.fa
```

(Hint: If we also included the quality string field in column 11, we could get fastq-files in the same way!)

8. Now that we have a fasta file, you can send it (or first download it to your computer if you are using GitHub codespaces), and send it to Blast.

   Select “Nucleotide blast”, and upload the fasta file via the “Choose file” button.

   Were the results as you expected?

More things you can do with samtools

Extracting unclassified reads

There are many more things you can do with samtools.

For example, to extract unclassified reads, you can use:

```
samtools view -f 4 <samfile> | less -S
```

… or even output fastq directly:

```
samtools fastq -f 4 <sam-file> > unclassified.fq
```

Note though that Emu also has an option to output unclassified reads in a separate file directly, using the `--output-unclassified` flag. See more info about that in the Emu README file