Training in genomic Epidemiology and Public Health Bioinformatics "Bridging the Gap"
Day 9 - Introduction to sequence databases and data sharing - practical exercises.

Objectives:

- Become familiar with the ENA submission portal.
- Successfully upload data to the ENA TEST database using the interactive route and learn about the programmatic method.
- Search and download data from ENA using the browser and command-line

The exercise consists of two parts: uploading and retrieving data to ENA. Please focus on the mandatory exercises and try the optional ones if you have time left at the end of the session. Optional exercises are intended to provide practical examples of the different submission and retrieval routes offered by ENA.

Pre-course activities

72 hours prior to the exercise, each participant should have created its own submission account by following this link: https://www.ebi.ac.uk/ena/submit/webin/accountInfo

The data used in the exercise is located in the following location:
BTG_2024/data/test_upload

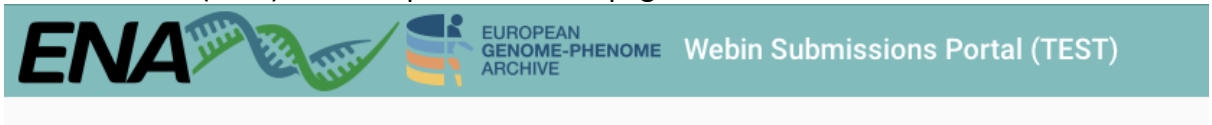
PART 1: Uploading raw reads to ENA

Exercise 1: Submitting raw reads through the interactive route

Step 1: Create a new project

Log on to the TEST Webin portal.
https://wwwdev.ebi.ac.uk/ena/submit/webin/login

It should show (TEST) on the top ribbon of the page:



1- Click on 'Register Study' and fill in the information. The creation requires you to set a 'Release date'. Pick one in the calendar that is different from today's date (the project will not persist more than 24 hours on the test server but let us set a date knowingly. Every data submitter should be aware that he has control on when the data shall be released). Add a short title and description and fill in the abstract field.

2- Submit the project. If you have created your submission account more than 72 hours prior to the exercise, you will receive a successful project creation pop-up message. In the project creation message, there will be a PREJBxxxx accession ID, write this down as it will be needed for the submission exercise.

In case the project had been erroneously created in the production server, you will have to send a service ticket to ENA with the study accession ID to request its removal. To do this, click on "Support" and follow instructions. Specify that you are the "account holder" and "I have a query/issue" related to "submission".

Step 2: Add new samples to an existing project.

Samples can be added to a project programmatically or interactively. This exercise will show you the process using the interactive route.

1- Go to the Webin portal and click on 'Register Samples'

2- Download the correct spreadsheet corresponding to foodborne pathogen samples.
For information, all checklists are described in the following webpage:
https://www.ebi.ac.uk/ena/browser/checklists

3- Fill in the sample spreadsheet for using the provided information. The required fields are shown in green in the checklist page

Use the taxon id 1639 for Listeria monocytogenes.

Mandatory and Recommended fields with no data cannot be left blank (remove columns)

Save the file in csv or tsv format.

4- Upload spreadsheet to register samples by clicking on "Submit Completed Spreadsheet" and check the newly registered samples by clicking on "Sample Report"

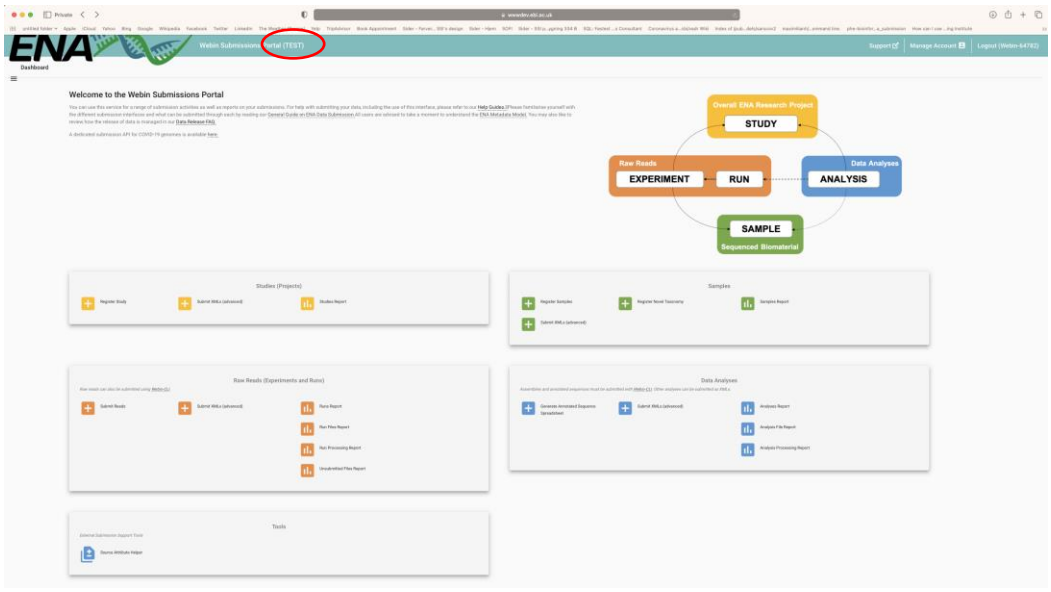
Step 3: Submit raw reads to an existing project and samples in the interactive way.

Detailed instructions for uploading files in the interactive way is available on the ENA webpage https://ena-docs.readthedocs.io/en/latest/submit/reads/interactive.html

In this exercise, we will add raw reads to samples that have just been created in step 2.

1- Go to the TEST submission portal (use your account credentials)

https://wwwdev.ebi.ac.uk/ena/submit/webin/login



2- Click on “Submit reads”.

The next page will suggest you download a template for the metadata to include in the submission. Click “Download spreadsheet for Read submission”.

You will be submitting paired-end Illumina raw reads using fastq files, find the correct tsv template and download it with the recommended fields.

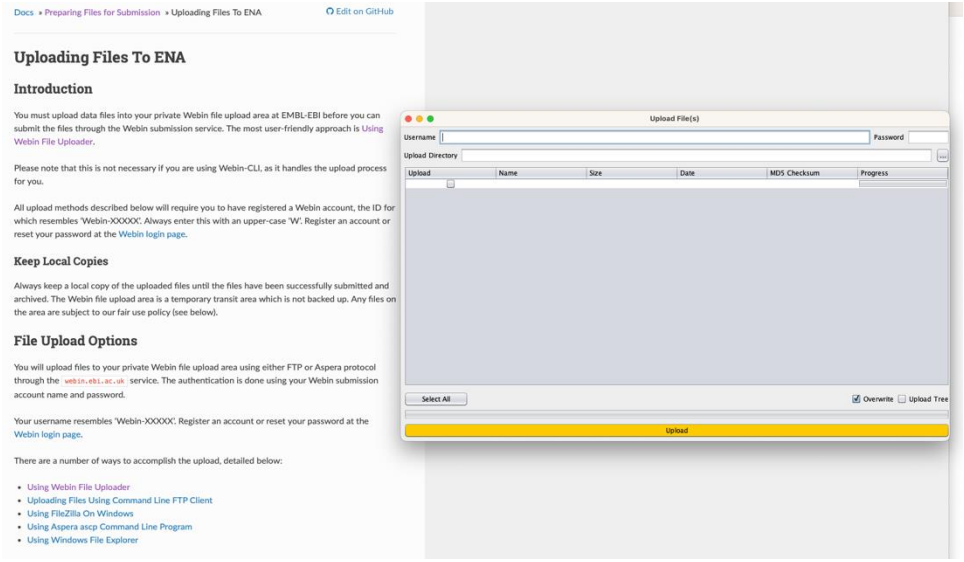
3- Check the “Sample report” and write down the sample accession ids SAMEAxxxxxx needed for the submission metadata sheet.

4- Fill in the downloaded template using the provided PRJEBxxxxx project identifier and metadata available metadata. Provide the file names for each fastq file and their matching md5sums. Save the file in csv or tsv format.

To calculate the md5sum of the fastq file, open a terminal and type the command:
md5sum strain0011_R*.fastq.gz

5- Now upload the raw reads fq.gz files matching the samples you included in the metadata sheet need to be uploaded to the test submission server using the ena file uploader.
Use the Webin uploader with your ENA login information

More information here https://ena-docs.readthedocs.io/en/latest/submit/fileprep/upload.html



In case of a network permission issue, fastq files can also be uploaded using the aspera tool (see programmatic raw read submission exercise Step 2)

6- Go back to the TEST submission portal and upload the filled spreadsheet. The file will be validated, and you will receive a pop-up message to confirm whether your raw read submission was successful or not.

7- Check your submission. Click on “Runs report” and check one of the records that you have just uploaded to ENA.

Optional exercises: programmatic raw read submission to ENA

This section provides you with some information to get started with doing larger submissions using the command line.
There are two command-line routes to submit data:

- The XML route
- The json-route

Programmatic submission using the json route can be done using the Webin REST API V1
https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit
The documentation is available here
https://ena-docs.readthedocs.io/en/latest/submit/general-guide/webin-v1.html

Json (and XML) submissions can be done through the new Webin REST V2 API.
https://wwwdev.ebi.ac.uk/ena/submit/webin-v2/
The documentation is available here:
https://ena-docs.readthedocs.io/en/latest/submit/general-guide/programmatic.html

Optional exercise 1: XML programmatic submission

Step 1: Add new samples to an existing project.

1- A sample submission XML must be prepared and submitted to the portal using the command-line.

An example of samplelist.xml file for the food pathogen test sample used in this exercise (checklist ERC00028) could be:

```
<?xml version="1.0" encoding="UTF-8"?>
<SAMPLE_SET>
 <SAMPLE alias=" strain0011_testsample_1">
  <TITLE>strain0011</TITLE>
  <SAMPLE_NAME>
   <TAXON_ID>1639</TAXON_ID>
   <SCIENTIFIC_NAME>Listeria monocytogenes</SCIENTIFIC_NAME>
  </SAMPLE_NAME>
  <DESCRIPTION>Original</DESCRIPTION>
  <SAMPLE_ATTRIBUTES>
   <SAMPLE_ATTRIBUTE>
    <TAG>collection date</TAG>
    <VALUE>2018</VALUE>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
    <TAG>geographic location (country and/or sea)</TAG>
    <VALUE>Denmark</VALUE>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
    <TAG>host health state</TAG>
    <VALUE>not provided</VALUE>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
    <TAG>host scientific name</TAG>
    <VALUE>Homo sapiens</VALUE>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
    <TAG>isolate</TAG>
    <VALUE> strain0011</VALUE>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
    <TAG>isolation source</TAG>
    <VALUE>blood</VALUE>
   </SAMPLE_ATTRIBUTE>
   <SAMPLE_ATTRIBUTE>
    <TAG>ENA-CHECKLIST</TAG>
    <VALUE>ERC000028</VALUE>
   </SAMPLE_ATTRIBUTE>
  </SAMPLE_ATTRIBUTES>
 </SAMPLE>
</SAMPLE_SET>
```

2- The command-line submission is done using a curl command and requires to attach a submission.xml file containing the lines shown below:

```
<?xml version="1.0" encoding="UTF-8"?>
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <ADD/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```

Run the submission command using your account credentials (make sure that the submission drop-box points to 'wwwdev.ebi.ac.uk).

curl -u Webin-XXXXX:yourpassword -F "SUBMISSION=@submission.xml" -F "SAMPLE=@samplelist.xml" https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit/ > sample_receipt.xml

Please make sure that you are using the correct environment. The server URL should start with **wwwdev.ebi.ac.uk**

3- After submission of the samples, you will receive an acknowledgement XML.
You will need to parse the sample accessions to submit data to the samples you have just created.

Step 2: Submit raw reads to an existing project and samples in the programmatic way.

If you have already submitted the raw read files in the first exercise using the interactive method, you must make a copy of the fastq.gz files and save the copy using a different name. ENA only allows raw read names with the same names to be submitted once.

Two XML files need to be created: an experiment XML which contains the information related to sequencing and a run XML containing the raw read file information.

An experiment XML looks as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<EXPERIMENT_SET><EXPERIMENT alias="test_experiment_1">
   <TITLE>test_experiment_1</TITLE>
   <STUDY_REF accession="PRJEBxxxxx" />
   <DESIGN>
    <DESIGN_DESCRIPTION/>
     <SAMPLE_DESCRIPTOR accession="SAMEAxxxxxxxx" />
     <LIBRARY_DESCRIPTOR>
       <LIBRARY_STRATEGY>WGS</LIBRARY_STRATEGY>
       <LIBRARY_SOURCE>GENOMIC</LIBRARY_SOURCE>
       <LIBRARY_SELECTION>RANDOM</LIBRARY_SELECTION>
       <LIBRARY_LAYOUT>
         <PAIRED NOMINAL_LENGTH="152"/>
       </LIBRARY_LAYOUT>
     </LIBRARY_DESCRIPTOR>
   </DESIGN>
   <PLATFORM>
     <ILLUMINA>
       <INSTRUMENT_MODEL>NextSeq 500</INSTRUMENT_MODEL>
     </ILLUMINA>
   </PLATFORM>
  </EXPERIMENT>
</EXPERIMENT_SET>
```

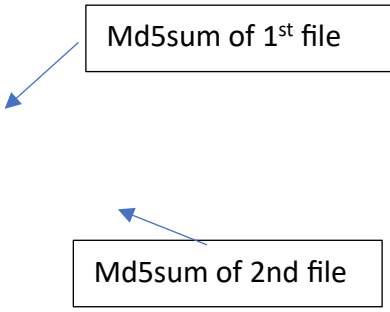
The project ID you created

The sample accession ID you received after step 1

A run XML looks as follows:

Must match the experiment alias

```
<?xml version="1.0" encoding="UTF-8"?>
<RUN_SET><RUN alias="test_1">
      <EXPERIMENT_REF refname="test_experiment_1"/>
      <DATA_BLOCK>
        <FILES>
          <FILE filename="strain0011_R1.fastq.gz" filetype="fastq"
            checksum_method="MD5" checksum="6f5c8df5c7c9962949a6b0c363f5c989"/>
          <FILE filename="strain0011_R2.fastq.gz" filetype="fastq"
            checksum_method="MD5" checksum="9f060bfee3f8c221c1e7d3d0ea647534"/>
        </FILES>
      </DATA_BLOCK>
   </RUN>
</RUN_SET>
```



Note: when you use the XML template, make sure to fill in the md5 checksum that matches the file that you submit.

IMPORTANT: Before submitting the XMLs, raw read files must be uploaded in advance using for example Aspera

Use the conda environment **BTG_aspera**

ascp -QT -l300M -L- <raw read file name> Webin-XXXXX@webin.ebi.ac.uk:.

or using the java uploader provided by ENA here as in the previous exercise:
https://ena-docs.readthedocs.io/en/latest/submit/fileprep/upload.html

After uploading the read files, submit the experiments and runs XMLs as follows:

curl -u Webin-XXXXX:yourpassword -F "SUBMISSION=@submission.xml" -F "EXPERIMENT=@experimentlist.xml" -F "RUN=@runlist.xml" https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit/ > exp_run_receipt.xml

Optional exercise 2 : json programmatic upload

This exercise will show you how to make a programmatic submission using json files.

Step1: Submit a sample.json file

A sample submission json must be prepared and submitted to the version 2 REST API.

An example sample.json file for the test data used in the exercise should look like as follows:

```
{
  "submission": {
    "alias": "submission5",
    "accession": "",
    "actions": [
      {
          "type": "ADD"
```

```
      },
      {
          "type": "HOLD",
          "holdUntilDate": "2025-01-12"
        }
    ]
  },
  "samples": [
    {
      "alias": "strain0011_testsample",
      "title": "strain0011",
      "organism": {
        "taxonId": "1639"
      },
      "attributes": [


        {
          "tag": "collection date",
          "value": "2018"
        },
        {
          "tag": "isolation source",
          "value": "blood"
        },
        {
          "tag": "human-associated environmental package",
          "value": "human-associated"
        },


        {
          "tag": "geographic location (country and/or sea)",
          "value": "Denmark"
        },
        {
          "tag": "host health state",
          "value": "not provided"
        },
        {
          "tag": "host scientific name",
          "value": "Homo sapiens"
        },

        {
          "tag": "isolate",
          "value": "strain0011"
        },
        {
          "tag": "ena-checklist",
          "value": "ERC000028"
        }
      ]
    }
  ]
}
```


You can submit the sample.json file also using the webin-cli instead of using the terminal and run the command above. To try it out go to this link, https://wwwdev.ebi.ac.uk/ena/submit/webin-v2/swagger-ui/index.html then authenticate using your account credentials. Click on submit, then try it out and attach the file to upload. Press the button 'execute' to run the command.

Step2: Submit an experiment and run .json file

Before submitting the json files, raw read files must be uploaded in advance using for example Aspera

conda activate BTG_aspera

ascp -QT -l300M -L- <raw read file name> Webin-XXXXX@webin.ebi.ac.uk:.

make a copy of the files using a different name from the first 2 submissions and use this name when you fill in the file information in the json.

An experiment and run json file for our example should look as follows:


```
{
  "submission":{
    "alias":"subs_test-alias-117",
    "accession":"",
    "actions":[
      {
        "type":"ADD"
      },
      {
        "type":"HOLD",
        "holdUntilDate":"2025-01-01"
      }
    ],
    "attributes":[
      {
        "tag":"test_tag",
        "value":"test_val"
      },
      {
        "tag":"test_tag_1",
        "value":"test_val_1"
      }
    ]
  },
  "runs":[
    {
      "alias":"run_alias_1",
      "identifiers":null,
      "centerName":"Statens Serum Institut",
      "title":"test",
      "description":"run for experiment alias 1",
      "experiment":{
        "alias":"experiment_alias_1"
      },
      "instrumentPlatform":"ILLUMINA",
      "instrumentModel":"NextSeq 500",
      "files":[
        {
          "fileName":"strain0011_R1.fastq.gz ",
          "fileType":"fastq",
          "checksumMethod":"MD5",
          "checksum":"6f5c8df5c7c9962949a6b0c363f5c989"
        },
```

```
                {
            "fileName":"strain0011_R2.fastq.gz ",
            "fileType":"fastq",
            "checksumMethod":"MD5",
            "checksum":"9f060bfee3f8c221c1e7d3d0ea647534"
          }
        ]
      }
    ],
    "experiments":[
      {
        "alias":"experiment_alias_1",
        "identifiers":null,
        "centerName":"Statens Serum Institut",
        "title":"Illumina NextSeq sequencing experiment 1",
        "study":{
          "accession":"PREJBxxxxx"
        },
        "samples":[
          {
            "accession":"SAMEAxxxxxx"
          }
        ],
        "designDescription":"",
        "libraryDescriptor":{
          "libraryStrategy":"WGS",
          "librarySource":"GENOMIC",
          "librarySelection":"RANDOM",
          "libraryLayout":"PAIRED"
        },
        "instrumentPlatform":"ILLUMINA",
        "instrumentModel":"NextSeq 500"
      }
    ]
}
```


The project ID you created

Sample id obtained from sample json submission

PART 2: Querying and retrieving data from ENA databases

Exercise 1: Use the ENA browser to search and retrieve data

The ENA browser is a free access resource for read sequence data. Datasets are organized as projects which could be searched using keywords.
Go to the ENA browser page https://www.ebi.ac.uk/ena/browser/home

a) Let us first try to find a record using its accession id.

Search the BioProject PRJEB25979. How many samples can you see? What type of data has been uploaded (e.g. reads, assemblies)?
Download the TSV summary of records and generate the script to fetch all the deposited files.
Try to select and download the files of the first record.

b) Now let us try the advanced search mode.

We want to search all SARS-CoV-2 (NCBI taxa 2697049) sequences from genomes collected in Denmark between August 1st, 2022 and August 5, 2022.

1- Go to Advanced search and create a query using the filters corresponding the criteria. Select all fields.

2- How many records did you find?

3- Download the full tsv report and all the fasta records.

You can also find step-by-step advanced search examples on the ENA documentation webpage here
https://ena-docs.readthedocs.io/en/latest/retrieval/advanced-search.html
You may try these if you have time.

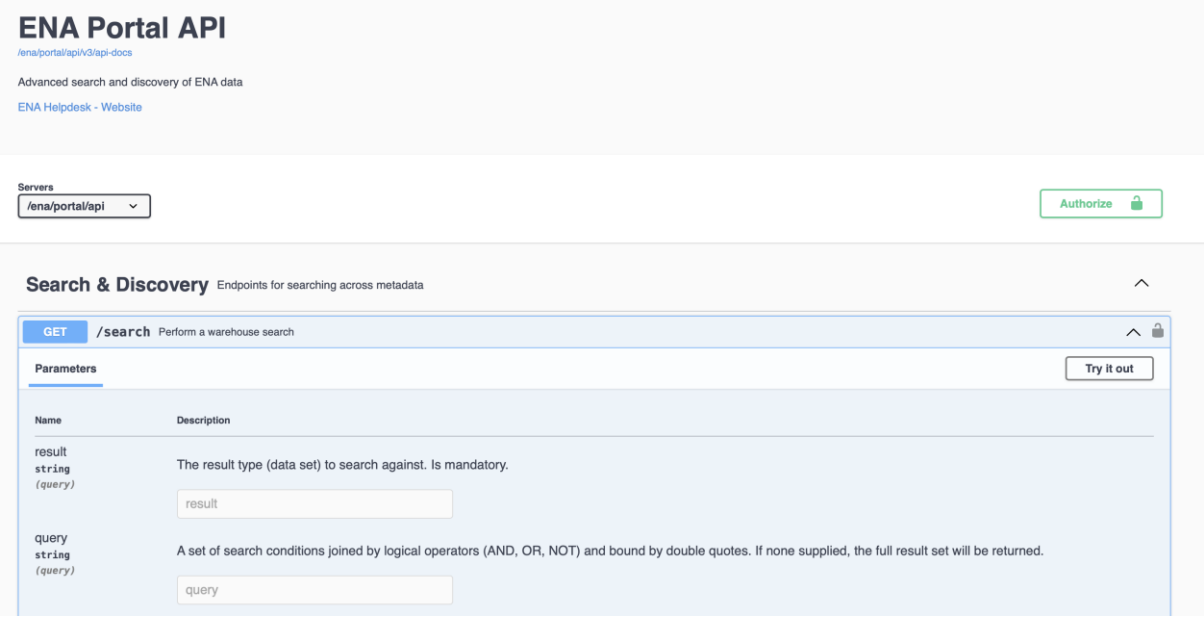
Exercise 2: Build a command line search query for the search API

Data queries can be done programmatically using the API. The full documentation is available here
https://docs.google.com/document/d/1CwoY84MuZ3SdKYocqssumghBF88PWxUZ/edit
And on ENA documentation webpage
https://ena-docs.readthedocs.io/en/latest/retrieval/programmatic-access/advanced-search.html

Write the query for the same search query as for the previous exercise (SARS-CoV-2 (NCBI taxa 2697049) sequences from genomes collected in Denmark between August 1st, 2022 and August 5, 2022)

Look at the available keywords for each criteria in the documentation.
We will use the search keywords **tax_id** (taxon id), **collection_date**, **country** and **sequence**.
Submit the query using the Webin API.
https://www.ebi.ac.uk/ena/portal/api/swagger-ui/index.html#/Search%20%26%20Discovery/search

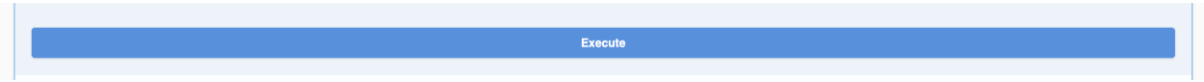
For search queries, authentication is not required (no need to click on “authorize”)



Click on 'try it out'

Use **sequence** as result type and write the search conditions in the query box

Click execute, the result will show below



The search query can also be submitted directly to the terminal without the interface. The search command is shown on the screen below the button 'execute' after submitting the query.

You may also try the example shown in the ENA documentation for a more complex query https://ena-docs.readthedocs.io/en/latest/retrieval/programmatic-access/advanced-search.html#retrieve-raw-read-and-primary-metagenome-datasets-for-cow-rumen-samples-collected-in-the-uk

Exercise 3a: Retrieving data files from ENA resources using the ena-file-downloader

Data files can be retrieved in different ways on ENA. The first exercised showed the ENA browser route.
This exercise will show you how to download data from the terminal using the java-based tool ena-file-downloader. The full documentation is shown here:
https://github.com/enasequence/ena-ftp-downloader/

Open a terminal and test the interactive mode of the java tool by just typing

java -jar ena-file-downloader.jar

Follow instructions and download files linked to the accession number: SAMEA1116772

Now try to run the tool in full command-line mode, the syntax is shown in the documentation

(Optional) Exercise 3b Retrieving data files from ENA resources using the ena-browser-tools

Optionally, you may try another command-line tool to retrieve data files from ENA. This tool is called enaDataGet, see documentation below.
https://ena-docs.readthedocs.io/en/latest/retrieval/programmatic-access/browser-tools.html

Use enaDataGet to download fastq files from this sample SAMEA1116773. An example of the command to run is shown in the documentation.

Exercise 4: Explore the ENA Pathogens portal

This exercise will allow you to explore the ENA Pathogen portal.
Go to https://www.pathogensportal.org

All data deposited on ENA associated to pathogens of interest can be searched using different entry points.

a) Let us first try “Sequences”. This search mode allows to find all sequences and raw reads for different pathogens.

Use the filters on the left columns to find all raw read datasets from Denmark for Listeria monocytogenes sequenced using Illumina.
How many samples do you see?

b) Go to “Outbreak”. Let us look at the data for mpxv (Monkeypox)

Use the filters in the left column to search all Monkeypox sequences. How many did you find? Look at the Nextstrain phylogenetic tree using the Nextstrain reports tab.

c) Go to Cohorts.

Not many cohort datasets are currently available on the Pathogens portal, only a few COVID-19 cohorts. Look at the different available data types. Clicking on the sequence data button will redirect you to the COVID-19 Data Portal.