training_modules Documentation

Release 1

Marc B Rossello

Jan 19, 2021
Welcome to the guidelines for submission and retrieval for the European Nucleotide Archive. Please use the links to find instructions specific to your needs. If you're completely new to ENA, you can see an introductory webinar at the bottom of the page.
Welcome to the general guide for the European Nucleotide Archive submission. Please take a moment to view this introduction and consider the options available to you before you begin your submission.

ENA allows submissions via three routes, each of which is appropriate for a different set of submission types. You may be required to use more than one in the process of submitting your data:

- **Interactive Submissions** are completed by either filling out web forms directly in your browser or alternatively downloading spreadsheets that can be completed off-line and uploaded to ENA. This is often the most accessible submission route.

- **Command Line Submissions** use our bespoke Webin-CLI program. This validates your submissions entirely before you complete them, allowing you maximum control of the process.

- **Programmatic Submissions** are completed by preparing your submissions as XML documents and either sending them to ENA using a program such as cURL or using the Webin Submissions Portal.

The table below outlines what can be submitted through each submission route. It is also recommended that you familiarise yourself with our metadata model.

<table>
<thead>
<tr>
<th></th>
<th>Interactive</th>
<th>Webin-CLI</th>
<th>Programmatic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Sample</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Read data</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Genome Assembly</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Transcriptome Assembly</td>
<td>N</td>
<td>Y</td>
<td>N</td>
</tr>
<tr>
<td>Template Sequence</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Other Analyses</td>
<td>N</td>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>

Before submitting to ENA it is important to familiarise yourself with the different options available for data submission. This general guide will get you started in understanding how data can be submitted to ENA and how to keep track of the progress of your submission.
1.1 Getting started

1.1.1 Register a Submission Account

Before you can submit data to ENA, you need to register for a Webin submission account. If you already have an account and have difficulties logging in please contact our helpdesk.

1. Navigate to the Interactive Webin submission service page.

2. Click the Register button.

3. Fill in the required details. Other contacts can be added after your account has been created.

4. Click the register button to register your account.

5. After registration you can log into Webin with your username and password. Please note that after registration you will be logged in automatically.

6. You can add more contacts by expanding My account details on the Home page.
1.1.2 The ENA Metadata Model

Submissions made through Webin are represented using a number of different metadata objects. Before submitting data to ENA, it is important to familiarise yourself with the ENA metadata model and what parts of your research project can be represented by which metadata objects. This will determine what you need to submit.

For example, a publication is typically associated with a study (project), sequenced source material is represented using samples, and sequencing experiment details are captured by the experiment object.

Note that data files are also submitted by associating them with metadata objects. Sequence read data is associated with run objects while other data files are associated with analysis objects. The full metadata model with relationships between the different types of objects is illustrated below.

Metadata Model

- **Study**: A study (project) groups together data submitted to the archive and controls its release date. A study accession is typically used when citing data submitted to ENA. Note that all associated data and other objects are made public when the study is released.
- **Sample**: A sample contains information about the sequenced source material. Samples are associated with checklists, which define the fields used to annotate the samples. Samples are always associated with a taxon.
- **Experiment**: An experiment contains information about a sequencing experiment including library and instrument details.
- **Run**: A run is part of an experiment and refers to data files containing sequence reads.
- **Analysis**: An analysis contains secondary analysis results derived from sequence reads (e.g. a genome assembly),
- **Submission**: A submission contains submission actions to be performed by the archive. A submission can add more objects to the archive, update already submitted objects or make objects publicly available.
1.1.3 Accession Numbers

Submissions to ENA result in accession numbers. A set of rules describe the format of the accessions, and these are described below, alongside examples of how they look. These accessions can be used to identify each unique part of your submission.

Please note, not all accessions become available in the browser and not all can be used in publications. For information on which accessions can be described in publications, see the guidelines at the bottom of this page.

Understanding these accessions can give you some information about what they refer to, even before you find them in our browser. For example, in the case of studies, samples, experiments, runs and analyses, you can identify which INSDC partner accepted the original submission by looking at the first letter: ‘E’ for ENA, ‘D’ for DDBJ, or ‘S’ for NCBI.

<table>
<thead>
<tr>
<th>Accession Type</th>
<th>Accession Format</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects</td>
<td>PRJ(E</td>
<td>D</td>
</tr>
<tr>
<td>Studies</td>
<td>(E</td>
<td>D</td>
</tr>
<tr>
<td>BioSamples</td>
<td>SAM(E</td>
<td>D</td>
</tr>
<tr>
<td>Samples</td>
<td>(E</td>
<td>D</td>
</tr>
<tr>
<td>Experiments</td>
<td>(E</td>
<td>D</td>
</tr>
<tr>
<td>Runs</td>
<td>(E</td>
<td>D</td>
</tr>
<tr>
<td>Analyses</td>
<td>(E</td>
<td>D</td>
</tr>
<tr>
<td>Assemblies</td>
<td>GCA_[0-9]{9}_[0-9]+</td>
<td>GCA_123456789.1</td>
</tr>
<tr>
<td>Assembled/Annotated Sequences</td>
<td>[A-Z][1]{0-9}[5].[0-9]+</td>
<td>A12345.1</td>
</tr>
<tr>
<td></td>
<td>[A-Z][2]{0-9}[6].[0-9]+</td>
<td>AB123456.1</td>
</tr>
<tr>
<td></td>
<td>[A-Z][3]{0-9}[8]</td>
<td>AB12345678</td>
</tr>
<tr>
<td></td>
<td>[A-Z][4]{0-9}[2]?[0-9]{6,8}</td>
<td>ABCD01123456</td>
</tr>
<tr>
<td></td>
<td>[A-Z][6]{0-9}[2]?[0-9]{7,9}</td>
<td>ABCDE011234567</td>
</tr>
<tr>
<td>Protein Coding Sequences</td>
<td>[A-Z][3]{0-9}[5].[0-9]+</td>
<td>ABC12345.1</td>
</tr>
<tr>
<td></td>
<td>[A-Z][3]{0-9}[7].[0-9]+</td>
<td>ABC1234567.1</td>
</tr>
</tbody>
</table>

How to cite your ENA study

In all cases, the top-level Project accession should be cited as well as a link to where the data can be found in the browser, for example:

“the data for this study have been deposited in the European Nucleotide Archive (ENA) at EMBL-EBI under accession number PRJEBxxxx (https://www.ebi.ac.uk/ena/browser/view/PRJEBxxxx).”

If there is a particular scenario where using the top level accession would not be suitable, for example, if you have multiple publications that reference individual components within a single ENA project (and therefore the project accession provides too much ambiguity), then the following accessions are also considered accessions that could be used for publication:

- Assemblies
- BioSamples (in the context of associated data)
- Assembled/Annotated Sequences
1.2 Submission services

1.2.1 Interactive Submission

Introduction

This guide will introduce you to submitting data interactively through your browser.

Production and Test Services

There are two interactive Webin submission services. One for test submissions and another for production submissions:

- Test service URL: https://wwwdev.ebi.ac.uk/ena/submit/sra
- Production service URL: https://www.ebi.ac.uk/ena/submit/sra

The test service is recreated from the full content of the production service every day at 03.00 GMT/BST. Therefore, any submissions made to the test service will be removed by the following day.

It is advisable to first test your submissions using the Webin test service before using the production service.

Submission

When you come to make a new submission, you will have the options described below. Follow the links for specific guidance on each submission activity. Please note that it is no longer possible to submit genome assemblies this way, please refer to the Webin-CLI interface for this.

1. Access the above image by logging into your Webin submission account and going to the ‘New Submission’ tab.

2. Use the Register study (project) option to register new studies.
3. Use the **Register samples** option to register new samples.

4. Use the **Submit sequence reads and experiments** option to submit sequence reads with associated experimental information. Ensure you have uploaded your read files first.

5. Interactive submission of genome assemblies is no longer possible. This must be done using Webin-CLI.

6. Use the **Submit other assembled and annotated sequences** option to submit coding genes, rRNA sequences and other annotated sequences.

7. Use the **Taxonomy Check/Request** option to check that your samples have recognised taxonomic classifications or to register new ones. You must provide a taxonomic classification for each submitted sample.

Each of these options will lead you to interactive web forms that guide you through the submission of these objects. Interactive submission is recommended for registration of your Study and Samples and for small scale Read or Sequence submission.

### 1.2.2 Programmatic Submission

**Introduction**

Submissions of different types can be made programmatically using XML.

A receipt XML with accession numbers is provided upon successful submission. Any validation or other errors will be written in the receipt XML as well.

**Production and test services**

There are two programmatic Webin submission services. One for test submissions and another for production submissions:

- **Test service URL:** https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit
- **Production service URL:** https://www.ebi.ac.uk/ena/submit/drop-box/submit

The test service is recreated from the full content of the production service every day at 03.00 GMT/BST. Therefore, any submissions made to the test service will be removed by the following day.

When you are using the test service the receipt XML will contain the following message:

```
<INFO>This submission is a TEST submission and will be discarded within 24 hours</INFO>
```

It is advisable to first test your submissions using the Webin test service before establishing an automated submission pipeline.

**Upload data files**

Data files must be uploaded into a submitter specific private Webin file upload area before they can be submitted. Always keep a local copy of the uploaded files until the files have been successfully submitted and archived. The Webin file upload area is a temporary transit area which is not backed up and subject to a fair use policy.

Once uploaded, data files are submitted by referring to them from Run XML or Analysis XML. Run XML is used for used for sequence read submissions while Analysis XML is used for other types of submissions (e.g. genome assemblies).

More information of uploading data files and our the fair use policy is available [here](#).
Submission protocol

Submissions are made through the secure HTTPS protocol using POST multipart/form-data according to RFC1867.

Authentication method

Webin user name and password must be provided using basic HTTP authentication.

When using curl the user name and password are provided using the `-u` option:

```
curl -u username:password
```

Types of XML

The type of each submitted XML file must be specified at time of submission.

When using curl each XML file is submitted using the `-F` option:

```
-F "XMLTYPE=@FILENAME"
```

where the XMLTYPE is one of the following POST parameters:

- SUBMISSION (XML Schema)
- STUDY (XML Schema)
- SAMPLE (XML Schema)
- EXPERIMENT (XML Schema)
- RUN (XML Schema)
- ANALYSIS (XML Schema)
- DAC (XML Schema)
- POLICY (XML Schema)
- DATASET (XML Schema)
- PROJECT (XML Schema)

Below is an example of a sequence read data submission to the Webin test service:

```
curl -u username:password -F "SUBMISSION=@submission.xml" -F "EXPERIMENT=@experiment.xml" -F "RUN=@run.xml" "https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit/"
```

Identifying objects

Each submitted object is uniquely identified within a submission account using the alias attribute. Once an object has been submitted no other object of the same type can use the same alias within the submission account. Objects can refer to other objects within a submission account by either alias or the assigned accession number.
Identifying submitters

The `center_name` attribute defines the submitting institution. The center name is automatically assigned from submission account details except for broker accounts. Brokers should provide a center name which reflects the institute where the data was generated. For brokers, the `broker_name` field is automatically assigned at time of submission.

Submission XML

Each submission should contain a submission XML file which defines the submission actions. The most commonly used submission actions are listed below.

Submission XML: submit new objects

The `ADD` action is used when submitting new objects. A corresponding submission XML will look like:

```
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <ADD/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```

Submission XML: submit studies with release date

If no release date is provided then submitted studies and any associated objects will be publicly released two months after the date of study submission.

A release date can be provided for studies by using the `HOLD` action together with the `ADD` action:

```
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <ADD/>
    </ACTION>
    <ACTION>
      <HOLD HoldUntilDate="TODO: release date"/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```

The `HoldUntilDate` specifies the public release date of any studies submitted within the submission. This can be at most two years in the future.

Submission XML: make study public

A study can be made immediately public by using `RELEASE` action with the study accession number:

```
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <RELEASE>
        ...
      </RELEASE>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```
Submission XML: set study hold date

You can update the release date of a study by specifying its accession alongside a new release date:

```xml
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <HOLD target="TODO: study accession number" HoldUntilDate="TODO: YYYY-MM-DD"/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```

This applies only to non-public studies. It is not possible to suppress a public study by this method.

The new release date must be not more than two years in the future.

Submission XML: update existing objects

The MODIFY action is used when updating existing objects. A corresponding submission XML will look like:

```xml
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <MODIFY/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```

Checklist for preparing an object for update:

- The new XML for the object must either contain the original alias or the assigned accession.
- If you provide an alias then it must be given exactly as in the initial submission.
- If you provide an accession then it must be the accession number assigned to the object in the initial submission.
- Run XML objects can’t be updated to point to different data files.
- Analysis XML objects can’t be updated to point to different data files.

Below is an extract of an study XML which contains an accession number:

```xml
<STUDY accession="ERP000011">
  ...
</STUDY>
```

Below is an extract of an study XML which contains an alias:
Submission XML: validate objects

The VALIDATE action is used when validating a submission without actually creating or updating any objects. It can be used together with the ADD action (default):

```
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <ADD/>
    </ACTION>
    <ACTION>
      <VALIDATE/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```

Or with the MODIFY action:

```
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <MODIFY/>
    </ACTION>
    <ACTION>
      <VALIDATE/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```

Submission XML: cancel objects

Pre-publication private objects (with the exception of assemblies) can be cancelled by all submitters.

To cancel objects, you must use the CANCEL action in the submission XML.

The CANCEL action should point to the object that is being cancelled. Please note that the CANCEL action will be propagated from studies to all associated experiments and analyses, and from experiments to all associated runs.

Below is an example of a submission XML used to cancel objects:

```
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <CANCEL target="TODO: object accession number"/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```
Submission XML: request a receipt XML

The RECEIPT action can be used to request a receipt XML issued as a result of an ADD action that contains all the issued accession numbers. This is especially useful in cases where the submission request times out before it has as completed. The submission process will continue nevertheless and the issued accession numbers can be later requested using the RECEIPT action.

```
SUBMISSION_SET>
   <SUBMISSION>
      <ACTIONS>
          <ACTION>
              <RECEIPT target="submission alias or accessions"/>
          </ACTION>
      </ACTIONS>
   </SUBMISSION>
</SUBMISSION_SET>
```

Submission actions without submission XML

Some submission actions can be defined using the ACTION POST parameter. If the ACTION parameter is provided then the submission XML will be ignored.

You can use the following actions in place of submission XML:

- ACTION=ADD: create new objects in the archive
- ACTION=MODIFY: update existing objects in the archive
- ACTION=VALIDATE (same as ACTION=VALIDATE,ADD)
- ACTION=VALIDATE,ADD: validate new objects without creating them
- ACTION=VALIDATE,MODIFY: validate update of existing objects without updating them

In addition, you can use the following POST parameters with the ACTION parameter:

- HOLD_DATE: set the public release date for a new study or sample
- CENTER_NAME: set the submitting center name (mandatory for brokers)

For example, submit a new analysis:

```
curl -u username:password -F "ACTION=ADD" "https://www.ebi.ac.uk/ena/submit/drop-box/submit/" -F "ANALYSIS=@analysis.xml"
```

or submit a new study with a public release date:

```
curl -u username:password -F "HOLD_DATE=31-11-2019" -F "ACTION=ADD" "https://www.ebi.ac.uk/ena/submit/drop-box/submit/" -F "PROJECT=@project.xml"
```

Receipt XML

Once a submission has been processed a receipt XML (XML Schema) is returned.

To know if the submission was successful look in the first line of the <RECEIPT> block.

The attribute success will have value true or false. If the value is false then the submission did not succeed. In this case check the rest of the receipt for error messages and after making corrections, try the submission again.
If the success attribute is true then the submission was successful. The receipt will contain the accession numbers of
the objects that you have submitted. In the case of a study submission this is likely to be the accession that you will be
including in a publication.

An example of a successful run submission:

```xml
<RECEIPT receiptDate="2014-12-02T16:06:20.871Z" success="true">
  <RUN accession="ERR049536" alias="run_1" status="PRIVATE"/>
  <SUBMISSION accession="ERA390457" alias="submission_1"/>
  <ACTIONS>ADD</ACTIONS>
</RECEIPT>
```

Above, the assigned run accession number ERR049536 is provided in the accession attribute within the RUN block.

If the submission was not successful the Receipt XML will contain the error messages within the MESSAGES block:

```xml
<RECEIPT receiptDate="2014-12-02T16:06:20.871Z" success="false">
  ...
  <MESSAGES>
    <ERROR>This is an error message.</ERROR>
  </MESSAGES>
  ...
</RECEIPT>
```

Retrieving submitted XMLs

Submitted XMLs can be retrieved from the submission service using the assigned accession numbers. This can be a
helpful, for example, as a starting point for making modifications to the submitted objects.

Please substitute `<accession>` with the assigned accession number in the URLs below:

- Project XML: https://www.ebi.ac.uk/ena/submit/drop-box/projects/<accession>
- Study XML: https://www.ebi.ac.uk/ena/submit/drop-box/studies/<accession>
- Sample XML: https://www.ebi.ac.uk/ena/submit/drop-box/samples/<accession>
- Run XML: https://www.ebi.ac.uk/ena/submit/drop-box/runs/<accession>
- Experiment XML: https://www.ebi.ac.uk/ena/submit/drop-box/experiments/<accession>
- Analysis XML: https://www.ebi.ac.uk/ena/submit/drop-box/analyses/<accession>

1.2.3 Webin-CLI Submission

Introduction

Submissions to ENA can be made using the interactive Webin submission service, programmatic Webin submission
service and the Webin command line submission service.

This module gives an introduction to the Webin command line submission interface used to validate, upload and
submit files to the European Nucleotide Archive (ENA) and will also link to where you can download it. Please note
that unlike with other ENA submissions routes you may have used, you do not need to pre-upload your files when
using Webin-CLI.

Webin-CLI is the only way to submit assembled genomes and transcriptomes.
Webin-CLI is available as a Docker image and as a Java jar file.

Download the program as a Java jar file

You can download Webin-CLI Java jar file from its GitHub repository. We recommend always using the latest version:

• Latest Webin-CLI Version

To get started with running Webin-CLI, download the .jar file for whatever version you’re interested in. If you have a GitHub account, you can use the ‘Watch’ button in the top right to always be notified of new releases.

Please note that Webin-CLI requires that you have Java installed before you can run it. You should have version 1.8 or newer installed, which can be downloaded from Java:

• Download Java

Webin-CLI has been tested against openjdk version 1.8.0_212. You are recommended to use equivalent or later version.

Download openapi JDKs from the below links:
https://adoptopenjdk.net/?variant=openjdk8&jvmVariant=hotspot

Download Oracle JREs from the below links:

• Latest JRE Version
• Older JRE Versions

Run the program as a Java jar file

The Webin command line submission interface is a self-executing Java jar file and is run using the java command:

```
java -jar webin-cli-<version>.jar <options>
```

for example:

```
java -jar webin-cli-1.7.3.jar <options>
```

The <version> is the version number of the program.

Please note that the command must include the location of the jar file. For example, if you have it in your Downloads directory, the appropriate command on Mac/Linux on immediately opening the terminal would be:

```
java -jar Downloads/webin-cli-1.7.3.jar <options>
```

On Windows a backward slash is used instead of a forward slash:

```
java -jar Downloads\webin-cli-1.7.3.jar <options>
```

The command line <options> are explained below.

Video Guide: Getting Started With Webin-CLI in Windows 10

Command Line Options

• -context: the submission type:
  - -context genome
  - -context transcriptome
- `--context sequence`
- `--context reads`

- `--userName`: the Webin submission account name.
- `--password`: the Webin submission account password.
- `--centerName`: the center name of the submitter (mandatory for broker accounts).
- `--manifest`: the manifest file name.
- `--outputDir`: directory for output files.
- `--inputDir`: input directory for files declared in manifest file.
- `--validate`: validates the files defined in the manifest file.
- `--submit`: validates and submits the files defined in the manifest file.
- `--test`: use Webin test service instead of the production service. Please note that the Webin upload area is shared between test and production services, and that test submission files will not be archived.
- `--ascp`: use Aspera Cli instead of FTP file transfer, if available. Aspera Cli should be installed and path to executable “ascp” should be in PATH variable.
- `--version`: prints the version number of the program and exists.
- `--help`: detailed information about the different options.

**Submission Process**

Please note that this section serves as a general overview of the use of Webin-CLI. You may prefer to find the page specific to your submission type using the links in the sidebar of this page. The following types of submissions are supported:

- genome assemblies
- transcriptome assemblies
- annotated sequences
- reads

The type of the submission is specified using the `--context` command line option:

- `--context genome`
- `--context transcriptome`
- `--context sequence`
- `--context reads`

The following picture illustrates the stages of the submission process:
Stage 1: Pre-register Study and Sample

Each submission must be associated with a pre-registered study and a sample.

- Register a Study
- Register a Sample

Stage 2: Prepare the Files

The set of files that are part of the submission must be specified using a manifest file. The manifest file is specified using the `--manifest <filename>` option.

Manifest File Format

The manifest file has two columns separated by a tab (or any whitespace characters):

- Field name (first column): case insensitive field name
- Field value (second column): field value

The manifest file contains metadata fields and file name fields.

Examples of metadata fields are study and sample references:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>STUDY</td>
<td>Study accession or unique name (alias)</td>
</tr>
<tr>
<td>SAMPLE</td>
<td>Sample accession or unique name (alias)</td>
</tr>
<tr>
<td>ANALYSIS_REF</td>
<td>Comma separated list of analysis accession(s)</td>
</tr>
<tr>
<td>RUN_REF</td>
<td>Comma separated list of run accession(s)</td>
</tr>
</tbody>
</table>

The file name field format is:

```
<file type>  <file name>
```
An example of a file name field is:

| FASTA genome.fasta.gz |

For example, the following manifest file represents a genome assembly consisting of contigs provided in one fasta file:

| STUDY TODO |
| SAMPLE TODO |
| ASSEMBLYNAME TODO |
| COVERAGE TODO |
| PROGRAM TODO |
| PLATFORM TODO |
| MINGAPLENGTH TODO |
| MOLECULETYPE genomic DNA |
| FASTA genome.fasta.gz |

**Manifest File Types**

Please refer to the more detailed documentation for supported file types for each submission.

Sequence based submission support the following formats:

- FASTA: Sequences in fasta format
- FLATFILE: Sequences in EMBL-Bank flat file format

The following additional formats are supported for **genome assembly submissions**:

- AGP: Sequences in AGP format
- CHROMOSOME_LIST: list of chromosomes
- UNLOCALISED_LIST: list of unlocalised sequences

The following formats are supported for **read submissions**:

- BAM: BAM file
- CRAM: CRAM file
- FASTQ: fastq file

**Info File (for backward compatibility only)**

You can also provide the metadata fields in a separate info file. The info file has the same format as the manifest file. When a separate info file is used then the manifest file must contain the **INFO** field pointing to the info file.

For example, the following manifest file represents a genome assembly consisting of contigs provided in one fasta file:

| INFO assembly.info |
| FASTA genome.fasta.gz |

**Stage 3: Validate and Submit Files**

You can validate your files using the **-validate** command line option. Note that the **-submit** option must be used to submit the validated files.
You can submit your files using the `submit` command line option. Before being submitted your files will be validated and uploaded to your private Webin file upload area in webin.ebi.ac.uk.

Please refer to individual modules for validation rules.

Validation error reports are written into the `<outputDir>/<context>/<name>/validate` directory.

The `webin` command line submission interface creates and submits XMLs for you. These XMLs and the Receipt XML containing accession numbers are written into the `<outputDir>/<context>/<name>/submit` directory. This directory also contains the file manifest that refers to the files that are part of the submission.

The `<outputDir>` can be specified using the `-outputDir` option, the `<context>` is specified using the `-context` option, and the `<name>` is a submitter provided unique name specified in the manifest file.

Once the submission is complete an accession number is immediately returned to the submitter by the `webin` command line submission interface. Please refer to individual modules for advice which long term stable accession numbers can be used in publications.

**Output Directory Structure**

An output directory can be specific to the `webin` command line submission interface using the `-outputDir` option. This directory will have the following subdirectories:

- `<context>/<name>/validate`
- `<context>/<name>/submit`

If the `-outputDir` option is not specified then the directory in which the `-manifest` file is used as the output directory.

The `<context>` is the submission type provided using the `-context` option and the `<name>` is the unique name provided in the manifest file.

- The `validate` directory contains the validation reports created using the `-validate` option.
- The `submit` directory contains the XMLs created by the `-submit` option including the Receipt XML. This directory also contains the file manifest that refers to the files that are part of the submission.

**Validation Reports**

If the `-validate` action fails for any reason then validation reports are written into directory:

 `<context>/<name>/validate`

The validation reports correspond to the input files with an added suffix `.report`.

For example, a validated fasta file `assembly.fasta` will have a corresponding validation report `assembly.fasta.report`.

Messages which can’t be attributed to a specific input file will be written to both standard out and in the following file:

 `<context>/<name>/validate/webin-cli.report`

**Run the program using the Docker image**

Webin-CLI is available as the `enasequence/webin-cli` Docker image.

You can run the Webin-CLI docker image using `docker`:

docker run enasequence/webin-cli
or using singularity:

```
singularity run docker://enasequence/webin-cli
```

The required command line options are explained below. Please remember to mount local directories containing the files to submit so that they available to the running container.

**Configuring Your Firewall For ENA Upload**

Some users may encounter problems connecting to the ENA FTP service, a necessary step in Webin-CLI submission. A possible solution to this is to ensure that your firewall is configured appropriately to allow you to connect to this service:

- Configuring Your Firewall For ENA Upload

**Proxy Servers**

If your organisation uses a https proxy you can set the following Java properties to instruct the webin-cli to use them:

- https.proxyHost
- https.proxyPort

For example:

```
java -Dhttps.proxyHost=proxy.com -Dhttps.proxyPort=8080 -jar webin-cli-<version>.jar
```

Similarly, if your organisation uses a ftp proxy you can set the following properties:

- ftp.proxyHost
- ftp.proxyPort

For example:

```
java -Dftp.proxyHost=proxy.com -Dftp.proxyPort=8080 -jar webin-cli-<version>.jar
```

**Using Aspera Instead of FTP to Upload Files**

By default the Webin command line interface will use FTP to upload files to the webin.ebi.ac.uk server. Alternatively, you may use the Aspera protocol by installing Aspera Cli and specifying the `-ascp` option.

Please note that that the folder containing the `ascp` command line program must be included in the PATH variable. The `ascp` command can be found from the `cli/bin` directory of the downloaded and expanded Aspera Cli archive file.

**Release policy**

- Webin-CLI uses standard three number semantic versioning.

- Patch releases: the third digit is incremented by one after backward compatible bug fixes. For example, from 1.0.1 to 1.0.2.

- Minor releases: the second digit is incremented by one after backward compatible new features. For example, from 1.0.1 to 1.1.0.
• Major releases: changes that break backward compatibility result in the first digit being incremented. For example, from 1.0.1 to 2.0.0.

• The definition of Webin-CLI backward compatibility is that there are no breaking changes to the command line usage or to the file formats.

• All releases are made immediately after bugs have been fixed or new features have been added.

• Releases are downloadable from: https://github.com/enasequence/webin-cli/releases.

• After all releases, we will endeavour to contact affected submitters who previously were unable to complete their submissions.

• Minor and Major releases will be announced to ena-announce@ebi.ac.uk mailing list.

• After Minor or Major releases, submitters will be asked to migrate to use this (or higher) version after a transition period.

• After a Minor release, we will give two weeks notice for submitters to migrate to the new (or higher) version.

• After a Major release, we will give at least two months notice for submitters to migrate to the new (or higher) version.

1.3 Additional Services

1.3.1 Webin Submissions Portal

Webin submissions portal is a simple interactive web interface which supports:

• Submission of XML metadata objects

• Update of XML metadata objects

• Reports from the Webin reports service

• Download of spreadsheet template for annotated sequences

Service URL

Production service URL:
https://www.ebi.ac.uk/ena/submit/webin/

Test service URL:
https://wwwdev.ebi.ac.uk/ena/submit/webin/

Authentication

Please use your Webin submission account credentials to login into the service.
Submit XMLs

Once you have logged in you can use the default page to submit XMLs to Webin.

In the example below, one or more studies are being submitted using a project XML (`project.xml`) and a submission XML (`sub.xml`). The project XML contains the study details while the submission XML defines the submission actions.
To submit, simply browse to the `project.xml` and `sub.xml` files and click submit.

You can see the receipt in the browser window either in a tabular format:

```
<table>
<thead>
<tr>
<th>Type</th>
<th>Accession</th>
<th>Unique name (alias)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project</td>
<td>PRJEB25022</td>
<td>cheddar_cheese</td>
</tr>
<tr>
<td>Submission</td>
<td>ERA1210181</td>
<td>SUBMISSION-16-02-2018-17:11:39:627</td>
</tr>
</tbody>
</table>
```

or you can look at the receipt XML:

```
<RECEIPT receiptDate="2018-02-16T17:11:40.637Z" submissionFile="sub.xml" success="true">
  <PROJECT accession="PRJEB25022" alias="cheddar_cheese" status="PRIVATE" holdUntilDate="2020-02-16Z"/>
  <SUBMISSION accession="ERA1210181" alias="SUBMISSION-16-02-2018-17:11:39:627"/>
  <MESSAGES>
    <INFO>Submission has been committed.</INFO>
    <INFO>This submission is a TEST submission and will be discarded within 24 hours</INFO>
  </MESSAGES>
  <ACTIONS>404</ACTIONS>
</RECEIPT>
```

### Reports

**1.3. Additional Services**
Studies

The Studies report lists all studies submitted through your submission account.

<table>
<thead>
<tr>
<th>Submit XML</th>
<th>Studies</th>
<th>Samples</th>
<th>Runs</th>
<th>Analyses</th>
<th>Run files</th>
<th>Analysis files</th>
<th>Run process</th>
<th>Analysis process</th>
<th>Unsubmitted files</th>
</tr>
</thead>
</table>

Shows submitted studies and their release statuses. Search by accession or unique name, or simply click search to show most recent submissions. The results will show the most recently submitted studies in your submission account.

Please click search to see the results.

<table>
<thead>
<tr>
<th>Accession or Name</th>
<th>Release status</th>
<th>Search</th>
<th>Reset</th>
<th>Maximum rows</th>
<th>Show unique name</th>
<th>Download all results</th>
</tr>
</thead>
</table>

The Release date shows when the study and any associated data will be made public. If the Release date is in the future then the Status will be Private. The status will become Public once the the study release date expires.

In the above example, the status of the study is Cancelled. This means that that study has been withdrawn before making it public. Studies that are withdrawn after they have become public have either Suppressed or Killed status.

Samples

The Samples report lists all samples submitted through your submission account.

<table>
<thead>
<tr>
<th>Submit XML</th>
<th>Studies</th>
<th>Samples</th>
<th>Runs</th>
<th>Analyses</th>
<th>Run files</th>
<th>Analysis files</th>
<th>Run process</th>
<th>Analysis process</th>
<th>Unsubmitted files</th>
</tr>
</thead>
</table>

Shows submitted samples and their release statuses. Search by accession or unique name, or simply click search to show most recent submissions. The results will show the most recently submitted samples in your submission account.

Please click search to see the results.

<table>
<thead>
<tr>
<th>Accession or Name</th>
<th>Release status</th>
<th>Search</th>
<th>Reset</th>
<th>Maximum rows</th>
<th>Show unique name</th>
<th>Download all results</th>
</tr>
</thead>
</table>

Like studies all samples have a Status. Note that samples are automatically made public when data referring to them is made public.

Runs

The Runs report lists all sequence read experiments submitted through your submission account.
Sequence read experiments are defined using run and experiment metadata objects. Data files are associated with runs while experiments link runs to studies and samples. Run and experiment are automatically made public when the study they are part of is made public.

**Analyses**

The **Analyses** report lists all analyses submitted through your submission account.

Analyses can be of many different types. Many analyses are automatically made public when the study they are part of is made public. Some analyses will always be kept private while their data is made available through other services.

**Run Files**

The **Run files** report lists all run files submitted through your submission account.

The **Archive status** column value becomes **File archived** once the file has been incorporated into the archive. Any pre-archival file validation errors are also shown in this column.
Analysis Files

The **Analysis files** report lists all analysis files submitted through your submission account.

<table>
<thead>
<tr>
<th>Submit XML</th>
<th>Studies</th>
<th>Samples</th>
<th>Runs</th>
<th>Analyses</th>
<th>Run files</th>
<th>Analysis files</th>
<th>Run process</th>
<th>Analysis process</th>
</tr>
</thead>
</table>

- Shows submitted analysis files and their archival status. Search by accession or other conditions, or simply click search to show most recent submissions. The results will show the most recently submitted analysis files in your submission account.

- **Accession**
- **File name**
- **File format**
- **File size**
- **MD5 checksum**
- **Archive status**
- **Action**

The **Archive status** column value becomes **File archived** once the file has been incorporated into the archive. Any pre-archival file validation errors are also shown in this column.

Run Process

The **Run process** report lists the processing status of runs submitted through your submission account.

<table>
<thead>
<tr>
<th>Submit XML</th>
<th>Studies</th>
<th>Samples</th>
<th>Runs</th>
<th>Analyses</th>
<th>Run files</th>
<th>Analysis files</th>
<th>Run process</th>
<th>Analysis process</th>
<th>Unsubmitted files</th>
</tr>
</thead>
</table>

- Shows the processing status of archived run files. Search by accession or other conditions, or simply click search to show most recent submissions. The results will show the processing status of archived run files in your submission account.

**Please click search to see the results.**

<table>
<thead>
<tr>
<th>Accession</th>
<th>Process status</th>
<th>Maximum rows</th>
<th>Download all results</th>
</tr>
</thead>
</table>

- **Accession**
- **Process status**
- **Process error**
- **Action**

- **ERR1352896** **COMPLETED**

When the **Process status** is **COMPLETED** then the run has been successfully processed by the archive. Note that submitted files may be archived and available for download without them being successfully processed. In this case derived file products, such as archive generated Fastq files, may be missing.

Analysis process

The **Analysis process** report lists the processing status of analyses submitted through your submission account.

<table>
<thead>
<tr>
<th>Submit XML</th>
<th>Studies</th>
<th>Samples</th>
<th>Runs</th>
<th>Analyses</th>
<th>Run files</th>
<th>Analysis files</th>
<th>Run process</th>
<th>Analysis process</th>
<th>Unsubmitted files</th>
</tr>
</thead>
</table>

- Shows the processing status of archived analysis files. Search by accession or other conditions, or simply click search to show most recent submissions. The results will show the processing status of archived analysis files in your submission account. The primary use of this report is to retrieve sequence accessions assigned as part of genome assembly, transcriptome assembly or sequence flat file submissions.

**Please click search to see the results.**

<table>
<thead>
<tr>
<th>Accession</th>
<th>Analysis type</th>
<th>Sequence accession</th>
<th>Process status</th>
<th>Process error</th>
<th>Action</th>
</tr>
</thead>
</table>

- **Accession**
- **Analysis type**
- **Sequence accession**
- **Process status**
- **Process error**
- **Action**

- **ERZ269236** **SEQUENCE ASSEMBLY** **CANCELLED**

When the **Process status** is **COMPLETED** then the analysis has been successfully processed by the archive. In the above example, the status of the analysis is **CANCELLED**. This means that analysis processing has been
cancelled and will no longer be attempted.

If any sequence accessions are assigned as part of the analysis processing then they will be listed in the **Sequence accession** column.

**Unsubmitted files**

Uploading a file to the Webin file upload area is not sufficient for inclusion into the archive. The file must also be submitted through the Webin submission service either as a run or analysis.

The Webin file upload area is a temporary transit area. Any unsubmitted files are subject to be removed according to ENA fair use policy.

The unsubmitted file report shows uploaded files scheduled to be removed unless submitted by the expiration date.

<table>
<thead>
<tr>
<th>Submit XML</th>
<th>Studies</th>
<th>Samples</th>
<th>Runs</th>
<th>Analyses</th>
<th>Run files</th>
<th>Analysis files</th>
<th>Run process</th>
<th>Analysis process</th>
<th>Unsubmitted files</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Note that unlike all other reports, the contents of this report may not be fully up to date. This report is refreshed at most once per day.*

<table>
<thead>
<tr>
<th>File name</th>
<th>File size</th>
<th>Upload date</th>
<th>Expiration date</th>
</tr>
</thead>
<tbody>
<tr>
<td>17310_2#.cram</td>
<td>47341473</td>
<td>20th Apr 2016</td>
<td>21st Jul 2016</td>
</tr>
<tr>
<td>17321_2#.cram</td>
<td>46614575</td>
<td>20th Apr 2016</td>
<td>21st Jul 2016</td>
</tr>
</tbody>
</table>

See the Webin submissions portal for more details.

**1.3.2 Webin Reports Service**

The Webin reports service is a collection of RESTful JSON endpoints used to retrieve information about metadata objects and data files submitted through the Webin submission services.

The service can be used to retrieve information about:

- Submitted metadata objects
- Metadata object statuses
- Submitted data files
- Data file archival statuses
- Run processing statuses
- Analysis processing statuses
- Assigned accessions

The reports can also be conveniently explored using the Webin submissions portal.

**Service URL**

Production service URL:

https://www.ebi.ac.uk/ena/submit/report

Test service URL:
https://wwwdev.ebi.ac.uk/ena/submit/report

**Authentication**

Authentication is done using basic HTTP authentication with Webin submission account credentials.

**Service Endpoints**

The service endpoints are documented using Swagger:

https://www.ebi.ac.uk/ena/submit/report/swagger-ui.html
CHAPTER 2

How to Register a Study

2.1 Introduction

Every data submission to ENA requires registration of a study object. This study is used to group other objects together, so it is the first step towards submitting your data to ENA.

The study and its associated data will not become public until the study release date has expired. The default for a newly registered study is to have a hold date two months after the submission date. You can change this date at will.
should you need to release the data or delay its publication further. However, you will no longer have this option once
the data are public. Read more on our data release policies FAQ.
Have a look at an example of a study in the ENA browser.

2.2 Accessions

Once a study is registered, Webin will report two accession numbers for the study. The first starts with PRJEB and
is called the BioProject accession. This is typically used in journal publications. The study will also be assigned an
alternative accession number that starts with ERP. This accession number is called the SRA (Sequence Read Archive)
study accession.
Always make a note of any accessions you receive as these are the unique identifiers for each of your submissions to
ENA.

2.3 Submission Options

Studies can be submitted one of two ways, using the Webin Interactive web interface or programmatically. For an
overview of these, please see the General Guide on Submitting to ENA.

2.3.1 Register a Study Interactively

This form is used to register a study (also referred to as a project). Studies are typically registered before any data
is submitted. Data can be added to the study at any time. Please see Interactive Submission Options to learn how to
access it. Fill it out and click ‘Submit’ to register your study and receive accession numbers. The details can be edited
and updated later, but you need to have registered a study to submit your data.

Locus Tag Prefixes

If you intend to submit an annotated genome assembly, you will need to register a locus tag prefix at this stage. In the
bottom right of the form, find the question about functional genome annotation and check the ‘Yes’ radio button and
enter your prefix into the box which appears. Once you submit this, it will take 24 hours for the prefix to be usable.

The prefix you enter must conform to the following rules:
• Starts with a letter
• Between 3 and 12 characters long
• Is upper case
• Contains only alpha-numeric characters and no symbols such as -_*

2.3.2 Register a Study Programmatically

The Study Object

A study (also referred to as a project) object is submitted in XML format like this:

```xml
<PROJECT_SET>
  <PROJECT alias="iranensis_wgs">
    <NAME>WGS Streptomyces iranensis</NAME>
    <TITLE>Whole-genome sequencing of Streptomyces iranensis</TITLE>
    <DESCRIPTION>The genome sequence of Streptomyces iranensis (DSM41954) was obtained using Illumina HiSeq2000. The genome was assembled using a hybrid assembly approach based on Velvet and Newbler. The resulting genome has been annotated with a specific focus on secondary metabolite gene clusters.</DESCRIPTION>
    <SUBMISSION_PROJECT>
      <SEQUENCING_PROJECT/>
    </SUBMISSION_PROJECT>
    <PROJECT_LINKS>
      <PROJECT_LINK>
        <XREF_LINK>
          <DB>PUBMED</DB>
          <ID>25035323</ID>
        </XREF_LINK>
      </PROJECT_LINK>
    </PROJECT_LINKS>
  </PROJECT>
</PROJECT_SET>
```

You can register one or more studies at the same time by using one `<PROJECT></PROJECT>` block for each study.

The study XML format is defined by the ENA.project.xsd XML schema. Studies can also be submitted using the SRA.study.xsd XML schema.

Create the Study XML

Below is an example XML for submitting a study. Change the XML by entering your own information and save it as a file, for example project.xml.

```xml
<PROJECT_SET>
  <PROJECT alias="cheddar_cheese">
    <TITLE>Characterisation of Microbial Diversity and Chemical Properties of Cheddar Cheese Prepared from Heat-treated Milk</TITLE>
    <DESCRIPTION>This study aimed to characterise the interaction of microbial diversity and chemical properties of Cheddar cheese after three different heat treatments of milk.</DESCRIPTION>
    <SUBMISSION_PROJECT>
      <SEQUENCING_PROJECT/>
    </SUBMISSION_PROJECT>
  </PROJECT>
</PROJECT_SET>
```

(continues on next page)

2.3. Submission Options 31
In your file `project.xml` change the value of `alias` to be a unique name. You may need this unique name to refer to your study when adding other objects to it. It can be a short acronym but it should be meaningful and memorable in some way.

Within the `<DESCRIPTION></DESCRIPTION>` block add an abstract detailing the project including any information that may be useful for someone to interpret your project correctly. Within the `<TITLE></TITLE>` block add a descriptive title.

**Create the Submission XML**

To submit a study or any other object(s), you need an accompanying submission XML in a separate file. Name this file `submission.xml`.

```xml
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <ADD/>
    </ACTION>
    <ACTION>
      <HOLD HoldUntilDate="TODO: release date"/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```

The submission XML declares one or more Webin submission service actions. In this case the first action is `<ADD/>` which is used to submit new objects. The study has a release date on which it will become public, along with all data submitted beneath it. By default this is two months after the data of submission. However, you can specify any date within two years of the present date. This is the purpose of the `<HOLD/>` action. The release date can be modified later by replacing `<ADD/>` with `<MODIFY/>` action.

The XMLs can be submitted programmatically, using CURL on command line or using the Webin submissions portal.

**Submit the XMLs using CURL**

CURL is a Linux/Unix command line program which you can use to send the `project.xml` and `submission.xml` to the Webin submission service.

```csh
curl -u username:password -F "SUBMISSION=@submission.xml" -F "PROJECT=@project.xml" "https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit/"
```

Please provide your Webin submission account credentials using the `username` and `password`.

After running the command above a receipt XML is returned. It will look like the one below:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="receipt.xsl"?>
<RECEIPT receiptDate="2017-05-09T16:58:08.634+01:00" submissionFile="submission.xml" success="true">
  <PROJECT accession="PRJEB20767" alias="cheddar_cheese" status="PRIVATE"/>
  <SUBMISSION accession="ERA912529" alias="cheese"/>
  <MESSAGES>
    </MESSAGES>
</RECEIPT>
```
Submit the XMLs using Webin submissions portal

XMLs can also be submitted interactively using the Webin submissions portal. Please refer to the Webin submissions portal document for an example how to submit a study using XML. Other types of XMLs can be submitted using the same approach.

The Receipt XML

To know if the submission was successful look in the first line of the <RECEIPT> block.

The attribute success will have value true or false. If the value is false then the submission did not succeed. In this case check the rest of the receipt for error messages and after making corrections, try the submission again.

If the success attribute is true then the submission was successful. The receipt will contain the accession numbers of the objects that you have submitted. In the case of an ENA study this is likely to be the accession that you will be including in a publication.

Accession numbers in the Receipt XML

Webin will report an accession number for the study that starts with PRJEB.

This accession number is called the BioProject accession and is typically used in journal publications. The study will also be assigned an alternative accession number that starts with ERP. This accession number is called the SRA (Sequence Read Archive) study accession.

Test and production services

Note the message in the receipt:

It is advisable to first test your submissions using the Webin test service where changes are not permanent and are erased every 24 hours.

Once you are happy with the result of the submission you can use the CURL command again but this time using the production service. Simply change the part in the URL from wwwdev.ebi.ac.uk to www.ebi.ac.uk:

Similarly, if you are using the Webin submissions portal change the URL from wwwdev.ebi.ac.uk to www.ebi.ac.uk.

2.3. Submission Options
3.1 Introduction

In order to submit sequencing or analysis data, it is important to first register your biological samples with ENA. Each sample in ENA represents biomaterial that a sequencing library was produced from.

A different sample should be registered for each sample used in the real-life sequencing project. These samples give context to your data so it is important to ensure the metadata associated with them is as extensive and accurate as possible.
3.1.1 Checklists

To ensure that each sample is registered with at least a minimum amount of metadata, ENA provides “Sample Checklists” which are used during registration to tailor these samples to fit minimum standards for different sample types.

These “Checklists” were developed in collaboration with different research communities to ensure that there is enough information provided for each type of sample that is deposited in ENA to comply with the FAIR data principles (making sure your associated data is Findable, Accessible, Interoperable and Reusable). Explore all our sample checklists here.

As a result, when registering a sample, it is important to choose the most relevant sample checklist available to you and provide at least the minimum metadata.

**Note:** If your sample metadata does not provide enough context for your data to be easily interpreted, you may be requested to update your samples, so it is important to take some time to choose the correct checklist and metadata when you first register your samples.

If you can not find a suitable checklist for your sample data type - please first contact our helpdesk for advice. Please only use the ENA default sample checklist if you have been advised that there are no other suitable options.

If you can not provide a value for a mandatory field within a checklist, please use one of the INDSC accepted terms for missing value reporting.

3.1.2 Taxonomy

Please also make sure you are familiar with the ENA’s taxonomy services and use the correct taxonomy to describe your samples.

In particular, consider the environmental taxonomy options available to you when working with environmental samples.

3.2 Accessions

Once the samples are registered, Webin will report two accession numbers for each sample. The first starts with SAMEA and is called the BioSample accession. The samples will also be assigned an alternative accession number that starts with ERS. This accession number is called the ENA sample accession.

Always make a note of any accessions you receive as these are the unique identifiers for each of your submissions to ENA.

3.3 Submission Options

Samples can be submitted one of two ways, using the Webin Interactive web interface or programmatically. For an overview of these, please see the General Guide on Submitting to ENA.

3.3.1 Register Samples Interactively

Samples represent the source material that a sequencing library was produced from. A sample may be connected to reads, assemblies and various types of interpreted data. Samples add context to the data associated with them, and good sample annotation makes your data more searchable.
All samples must conform to a checklist of values which depend on the type of sample. The easiest way to view the available checklists is the sample checklist portal. Here you can select a checklist, customise its values, and view the allowed input for each field. You can then download it in spreadsheet form and upload it as described in Step 1, below. However, if you prefer to fill in a web form to make your sample submission, please read the full article that follows.

There are three steps to registering your sample(s).

**Step 1: Select A Sample Checklist**

Please see Interactive Submission Options for how to access this form.

ENA provides sample checklists which define a set of mandatory and recommended attributes for different types of samples. By declaring that you would like to register your samples using a specific checklist you are enabling the samples to be validated for correctness at submission time and are making it easier for other services to find and access the sample attribute information.

1. Click ‘Select Checklist’ to move on to the next page and select a checklist.

2. If you have created your sample submission in a spreadsheet, click here to submit it, then go to Step 3. Otherwise, ignore this.

3. Find a checklist that suits your type of sample.

4. Move on to Step 2 to create the sample template.
Chapter 3. How to Register Samples

3. Environment Checklists
   This group currently includes Genomic Standards Consortium (GSC) MixS sample checklists

   - ENA UniEuk_EukBank Checklist
     Minimum information required for reporting samples associated with the UniEuk EukBank initiative.
     This checklist aims to capture contextual metadata associated with V4 18S SSU rRNA molecular data.

   - GSC MixS human gut
     Genomic Standards Consortium package extension for reporting of measurements and observations relevant subsets of the GSC terms.

   - GSC MixS human skin
     Genomic Standards Consortium package extension for reporting of measurements and observations relevant subsets of the GSC terms.

4. Marine Checklists
   This group currently includes Micro B3 and Tara Oceans sample checklists

4. Pathogens Checklists
   This group currently includes several prokaryote and virus pathogen sample checklists

4. Other Checklists
   This group currently includes the ENA default sample checklist and a few project specific checklists

Restart Submission
Step 2: Select The Checklist Fields

1. Here, you can view the attributes in your chosen checklist. You can see which are optional and which are mandatory.

2. You can add more attributes if there are important details not already covered by the checklist.

3. The checklist can be downloaded in a format which you can edit as a spreadsheet. We recommend this since it allows you to more easily register multiple samples in a single submission and is more durable than a web form. See Step 1 to submit your spreadsheet.

4. Click next when you have finished customising your checklist.
5. In this form, enter any details which will be the same for all your Samples. They can be edited individually later. Click next when you have finished.
Step 3: Edit And Submit Samples

This screen allows you to edit samples individually. It is also where you will be directed if you submit via spreadsheet.

1. You can add more samples to your submission here. They must use the same checklist.

2. All your samples are listed here. The unique name is shown, along with an icon to indicate whether it is valid or not. Click the red cross to delete a sample entirely.

3. Use these buttons to move between samples.

4. Provide descriptive names here. The Unique name cannot be changed after submission.

5. A green tick indicates that the field’s content is valid, while a red cross indicates it requires your attention. Hover over the exclamation mark to see what should be entered.

6. Click here to download all the data you have submitted so far. This is useful if you have to stop halfway through your submission.

7. When all samples are complete, click here to submit. Webin will deliver accessions.

The samples you have just registered are not yet associated with a study. When you submit data, you will refer it to a study and sample, thus indirectly linking the sample with a study. See our Metadata Model for more information.

3.3.2 Register Samples Programmatically

The Sample Object

A sample object is submitted in XML format like this:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<SAMPLE_SET>
  
</SAMPLE_SET>
```

(continues on next page)
<SAMPLE alias="MT5176" center_name="">
  <TITLE>human gastric microbiota, mucosal</TITLE>
  <SAMPLE_NAME>
    <TAXON_ID>1284369</TAXON_ID>
    <SCIENTIFIC_NAME>stomach metagenome</SCIENTIFIC_NAME>
    <COMMON_NAME></COMMON_NAME>
  </SAMPLE_NAME>
  <SAMPLE_ATTRIBUTES>
    <SAMPLE_ATTRIBUTE>
      <TAG>investigation type</TAG>
      <VALUE>mimarks-survey</VALUE>
    </SAMPLE_ATTRIBUTE>
    <SAMPLE_ATTRIBUTE>
      <TAG>sequencing method</TAG>
      <VALUE>pyrosequencing</VALUE>
    </SAMPLE_ATTRIBUTE>
    <SAMPLE_ATTRIBUTE>
      <TAG>collection date</TAG>
      <VALUE>2010</VALUE>
    </SAMPLE_ATTRIBUTE>
    <SAMPLE_ATTRIBUTE>
      <TAG>host body site</TAG>
      <VALUE>Mucosa of stomach</VALUE>
    </SAMPLE_ATTRIBUTE>
    <SAMPLE_ATTRIBUTE>
      <TAG>human-associated environmental package</TAG>
      <VALUE>human-associated</VALUE>
    </SAMPLE_ATTRIBUTE>
    <SAMPLE_ATTRIBUTE>
      <TAG>geographic location (latitude)</TAG>
      <VALUE>1.81</VALUE>
      <UNITS>DD</UNITS>
    </SAMPLE_ATTRIBUTE>
    <SAMPLE_ATTRIBUTE>
      <TAG>geographic location (longitude)</TAG>
      <VALUE>-78.76</VALUE>
      <UNITS>DD</UNITS>
    </SAMPLE_ATTRIBUTE>
    <SAMPLE_ATTRIBUTE>
      <TAG>geographic location (country and/or sea)</TAG>
      <VALUE>Colombia</VALUE>
    </SAMPLE_ATTRIBUTE>
    <SAMPLE_ATTRIBUTE>
      <TAG>geographic location (region and locality)</TAG>
      <VALUE>Tumaco</VALUE>
    </SAMPLE_ATTRIBUTE>
    <SAMPLE_ATTRIBUTE>
      <TAG>environment (biome)</TAG>
      <VALUE>coast</VALUE>
    </SAMPLE_ATTRIBUTE>
    <SAMPLE_ATTRIBUTE>
      <TAG>environment (feature)</TAG>
      <VALUE>human-associated habitat</VALUE>
    </SAMPLE_ATTRIBUTE>
    <SAMPLE_ATTRIBUTE>
      <TAG>environment (material)</TAG>
      <VALUE>gastric biopsy</VALUE>
    </SAMPLE_ATTRIBUTE>
  </SAMPLE_ATTRIBUTES>
</SAMPLE>
You can register one or more samples at the same time by using one <SAMPLE></SAMPLE> block for each sample. The sample XML format is defined by the SRA.sample.xsd XML schema.

Most of the sample information comes in the form of <TAG> and <VALUE> pairs that belong in <SAMPLE_ATTRIBUTE> blocks. You can have any number of <SAMPLE_ATTRIBUTE> blocks in your samples.

Most submitters will want to use attributes that are recognised by ENA as these are better indexed for searching and filtering. For this purpose, it is recommended that submitters use ENA sample checklist attributes whenever possible. You can also use a combination of your own attributes with those recognised by ENA.

The Sample Checklists

ENA provides sample checklists which define all the mandatory and recommended attributes for specific types of samples. By declaring that you would like to register your samples using a specific checklist you are enabling the samples to be validated for correctness at submission time and are making it easier for other services to find and access the sample attribute information.

The sample with the above SAMPLE_ATTRIBUTE will be validated using the checklist ERC000014. The checklist is defined using the ENA-CHECKLIST attribute.

Note that the checklist is defined using a SAMPLE_ATTRIBUTE block and that the checklist defines the other SAMPLE_ATTRIBUTE blocks.

If you do not define a checklist then the samples will be validated against the ENA default checklist ERC000011. This checklist has virtually no mandatory fields but contains many optional attributes that can help you to annotate your samples to the highest possible standard.

You can find all the sample checklists here. For example, the checklist ERC000014 represents the GSC MiXS annotation standard for human associated source samples.

The checklists are defined using XML. These XMLs are available by appending &display=xml to the URL for retrieving a specific checklist. For example, the XML for checklist ERC000014 can be retrieved using the following URL: https://www.ebi.ac.uk/ena/browser/api/xml/ERC000011

The Taxonomic Classification

Note the <SAMPLE_NAME> block from the example above:
You must provide the ID of the species-level taxon you wish to use (<TAXON_ID>). The scientific name and common name are optional and will automatically be filled in for you.

Taxon IDs are drawn from the NCBI Taxonomy database. If you aren’t sure which taxon you need to use, or aren’t sure of its taxon ID, you can learn how to find this out in the Tips for Sample Taxonomy FAQ page.

In the above example, the sample source is environmental (stomach metagenome) and represents an unknown variety and quantity of organisms. Note that metagenomes use specific environmental terms.

Create the Sample XML

Below is an example XML for submitting a sample. Change the XML by entering your own information and save it as a file, for example sample.xml.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<SAMPLE_SET>
  <SAMPLE alias="MT5176">
    <TITLE>human gastric microbiota, mucosal</TITLE>
    <SAMPLE_NAME>
      <TAXON_ID>1284369</TAXON_ID>
    </SAMPLE_NAME>
    <SAMPLE_ATTRIBUTES>
      <SAMPLE_ATTRIBUTE>
        <TAG>collection date</TAG>
        <VALUE>2010</VALUE>
      </SAMPLE_ATTRIBUTE>
    </SAMPLE_ATTRIBUTES>
  </SAMPLE>
</SAMPLE_SET>
```

Create the Submission XML

To submit a sample or any other object(s), you need an accompanying submission XML in a separate file. Let’s call this file submission.xml.

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

The submission XML declares one or more Webin submission service actions. In this case the action is <ADD/> which is used to submit new objects.

The XMLs can be submitted programmatically, using CURL on command line or using the Webin submissions portal.
**Submit the XMLs using CURL**

CURL is a Linux/Unix command line program which you can use to send the sample.xml and submission.xml to the Webin submission service.

```
curl -u username:password -F "SUBMISSION=@submission.xml" -F "SAMPLE=@sample.xml" -d "https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit/"
```

Please provide your Webin submission account credentials using the username and password.

After running the command above a receipt XML is returned. It will look like the one below:

```
<?xml version="1.0" encoding="UTF-8"?>
<?xml-stylesheet type="text/xsl" href="receipt.xsl"?>
<RECEIPT receiptDate="2017-07-25T16:07:50.248+01:00" submissionFile="submission.xml" success="true">
  <SAMPLE accession="ERS1833148" alias="MT5176" status="PRIVATE">
    <EXT_ID accession="SAMEA104174130" type="biosample"/>
  </SAMPLE>
  <SUBMISSION accession="ERA979927" alias="MT5176_submission"/>
  <MESSAGES>
    <INFO>This submission is a TEST submission and will be discarded within 24 hours</INFO>
  </MESSAGES>
  <ACTIONS>ADD</ACTIONS>
</RECEIPT>
```

**Submit the XMLs using Webin submissions portal**

XMLs can also be submitted interactively using the Webin submissions portal. Please refer to the Webin submissions portal document for an example how to submit a study using XML. Other types of XMLs can be submitted using the same approach.

**The Receipt XML**

To know if the submission was successful look in the first line of the `<RECEIPT>` block.

The attribute `success` will have value `true` or `false`. If the value is false then the submission did not succeed. In this case check the rest of the receipt for error messages and after making corrections, try the submission again.

If the success attribute is true then the submission was successful. The receipt will contain the accession numbers of the objects that you have submitted.

**Accession numbers in the Receipt XML**

Webin will report an accession number for the sample that starts with SAMEA.

```
<SAMPLE accession="ERS1833148" alias="MT5176" status="PRIVATE">
  <EXT_ID accession="SAMEA104174130" type="biosample"/>
</SAMPLE>
```

This accession number is called the BioSample accession and is typically used in journal publications. The sample will also be assigned an alternative accession number that starts with ERS. This accession number is called the SRA (Sequence Read Archive) sample accession.

3.3. Submission Options 45
Test and production services

Note the message in the receipt:

```xml
<INFO>This submission is a TEST submission and will be discarded within 24 hours</INFO>
```

It is advisable to first test your submissions using the Webin test service where changes are not permanent and are erased every 24 hours.

Once you are happy with the result of the submission you can use the CURL command again but this time using the production service. Simply change the part in the URL from `wwwdev.ebi.ac.uk` to `www.ebi.ac.uk`:

```bash
curl -u username:password -F "SUBMISSION=@submission.xml" -F "SAMPLE=@sample.xml" "https://www.ebi.ac.uk/ena/submit/drop-box/submit/"
```

Similarly, if you are using the Webin submissions portal change the URL from `wwwdev.ebi.ac.uk` to `www.ebi.ac.uk`.

All files that are submitted to ENA need to be in the appropriate format to be accepted.

In the case of any data submitted Interactively or Programmatically you will also need to register the md5 checksum of your files and upload these to your Webin Upload Area prior to submission.

If data is submitted using Webin-CLI there is no need to register md5 checksums or previously upload data as the tool will do all this for you at the time of submission.

Below, you can find guides on the formats available for read and assembly submissions, as well as information on the EMBL flat file format, with templates. These are followed by sections on general steps you may need to take to prepare your file to be uploaded, and finally, how to carry out the actual upload.

### 4.1 Accepted Read Data Formats

#### 4.1.1 Single cell read data

Single cell read data must be submitted in the **BAM** or **CRAM** format using the following tags specified in the **SAM Optional Fields Specification**:

- CB: Cell identifier
- CR: Cellular barcode sequence bases (uncorrected)
- CY: Phred quality of the cellular barcode sequence in the CR tag

#### 4.1.2 Other read data

We recommend that all read data is submitted in the **BAM** or **CRAM** format. However, please note that a variety of other data formats are supported as well.
4.1.3 Sample de-multiplexing

Reads for different samples should be submitted using separate files. The only exception is when a BAM or CRAM file contains reads for a large number of samples intended to be always analysed together. In this case the sample associated with the read file should describe the sample group while the BAM or CRAM file should identify the sample for each read.

4.1.4 Standard formats

The following standard file formats are accepted and transformed into Fastq products:

- cram
- bam
- fastq

CRAM format

Each submitted CRAM file must:

- be compatible with the CRAM Format Specification
- be readable with Samtools
- contain only reference sequences that exist in the CRAM Reference Registry
- be submitted as a separate run
- use the .cram file name suffix (e.g. ‘a.cram’)

CRAM file names are required to end up with the .cram suffix (e.g. ‘a.cram’).

A CRAM index (CRAI) file is created by the archive for each submitted CRAM file and is available in the same directory as the CRAM file from which it was created. CRAM index file names start with the CRAM file name and end up with the .crai suffix (e.g. ‘a.cram.crai’ for CRAM file ‘a.cram’).

BAM format

Each submitted BAM file must:

- be compatible with the SAM/BAM Format Specification
- be readable with Samtools
- be submitted as a separate run
- use the .bam file name suffix (e.g. ‘a.bam’)

PacBio BAM files

We support the submission of the following types of PacBio BAM files:

- subread BAM files (*.subreads.bam)
- CCS read BAM files (*.ccs.bam)
Fastq format

We recommend that read data is either submitted in BAM or CRAM format. However, single and paired reads are accepted as Fastq files that meet the following requirements:

- Quality scores must be in Phred scale.
- Both ASCII and space delimited decimal encoding of quality scores are supported. We will automatically detect the Phred quality offset of either 33 or 64.
- No technical reads (e.g. adapters, linkers, barcodes, primers) are allowed.
- Single reads must be submitted using a single Fastq file and can be submitted with or without read names.
- Paired reads must be submitted using two Fastq files.
- The first line for each read must start with ‘@’.
- The base calls and quality scores must be separated by a line starting with ‘+’.
- Paired read names must either use Casava 1.8 read names (regular expression:
  `^@\([a-zA-Z0-9-_.]+:\d+:\d+:\d+:\d+:\d+:\d+\):(\d+):[YN]:\d+\:[02468]:[ACGTN]+\)$`
  or must end with /1 or /2 optionally followed by a space and a comment.
- The Fastq files must be compressed using gzip or bzip2.
- The regular expression for bases is “^([ACGTNactgn.]*?)$”

Example of Fastq file containing single reads:

```
@read_name
GATTTGGGTTCAACAGTCATCTGATCAAATAGTGAATCCATTTGTTCAACTCAGTTT
+
!''*(%%++)($%).1.****+*+'
```

Example of Fastq file containing paired reads (prior to Casava 1.8):

```
@read_name/1
GATTTGGGTTCAACAGTCATCTGATCAAATAGTGAATCCATTTGTTCAACTCAGTTT
+
!''*(%%++)($%).1.****+*+'
@read_name/2
GATTTGGGTTCAACAGTCATCTGATCAAATAGTGAATCCATTTGTTCAACTCAGTTT
+
!''*(%%++)($%).1.****+*+'
```

With Casava 1.8 the format of the ‘@’ line has changed and we accept this pattern too:

```
@EAS139:136:FC706VJ:2:2104:15343:197393 1:Y:18:ATCACG
```

4.1.5 Platform specific formats

Oxford Nanopore

Oxford Nanopore native data must be submitted as a single tar.gz archive containing basecalled fast5 files from Guppy, Metrichor, or Albacore.
For Metrichor, an example directory structure for run named XYZ:

```
XYZ/reads/downloads/fail/
XYZ/reads/downloads/pass/
```

How to archive all files in the XYZ downloads directory in a linux command line:

```
cd <directory containing XYZ directory>
tar -cvzf XYZ.tar.gz XYZ/reads/downloads/
```

**PacBio**

PacBio data submissions are supported in the platform specific native format.

One run consists of *.bax.h5, *.bas.h5 and xml files. Please note that these files must not be tarred.

**SFF format**

The SFF format is supported for the 454 and Ion Torrent platforms.

**10x Genomics**

To submit 10x Genomics data where read indexes exist, you must convert to BAM or CRAM format. The supported tags are defined in the [SAM Optional Fields Specification](#).

### 4.1.6 Formats being deprecated

**Complete Genomics native format**

The full Complete Genomics data package can be submitted including the ASM, LIB and MAP subfolders. Each data package should be submitted as a single experiment and run. Please note the data package must not be tarred or gzipped for submission.

**SRF format**

The *_seq.txt files can be converted into SRF files using the illumina2srf utility available from the DNA Sequence Read Toolkit.

Each Illumina lane should be submitted as a separate SRF file and runs should be demultiplexed prior SRF file generation.

To produce a SRF submission file for a non-paired lane, change the working directory to the run folder and run:

```
illumina2srf -R -P -N <run>:%l:%t: -n %x:%y -o <center_name>_<run>_<lane>.srf s_<lane> ...

→ *_seq.txt
```

The -R, -P options are used to exclude intensity, noise and signal data from the generated SRF files. These data series are no longer supported for new data submissions.

The recommended format for the SRF file names is `<center_name>.srf`, where `<center_name>` is the center name abbreviation assigned to all submitters, and the and are the run and the lane identifiers.

To produce a SRF submission file for paired lane, change the working directory to the run folder and run:
4.1.7 Deprecated formats

Read submissions are no longer accepted in the following formats:

- SOLiD csfasta/qual format (support ended in 2015)
- Illumina qseq format (support ended in 2015)
- Illumina scarf format (support ended in 2015)

4.2 Accepted Genome Assembly Data Formats

4.2.1 Introduction

The advice here is appropriate for submission of complete or near-complete replicons, including plasmids, organelles, complete viral genomes, viral segments/replicons, bacteriophages, prokaryotic and eukaryotic genomes. Chromosomes include organelles (e.g. mitochondrion and chloroplast), plasmids and viral segments.

Genome assembly data files might contain:

- Contig sequences
- Scaffold sequences
- Chromosome sequences
- Unlocalised sequences
- Functional annotation

Submission of single records that represent all of the unplaced scaffolds, or all of the scaffolds that belong to a particular chromosome but are not localized to a specific position on the chromosome, are not accepted. These records do not represent biological objects and should therefore be split into individual records for each scaffold.

You can use the below file formats to submit genome assemblies. Follow the links to learn more about formatting them:

- **FASTA File**: Unannotated assemblies should be submitted as a FASTA file
- **Flat File**: Annotated assemblies must be submitted as an EMBL flat file
- **AGP File**: The assembly of scaffolds or chromosomes from contigs can be described using an AGP file
- **Chromosome List File**: Must be provided when the submission contains assembled chromosomes
- **Unlocalised List File**: Should be provided when the submission contains chromosomes with unlocalised sequences

Please note that all data files must be compressed with GZIP.

Some additional information is provided in the appendices:

- **Appendix: Unique Sequence Names**
- **Appendix: Definition of Terms**
4.2.2 FASTA File

Unannotated sequences should be submitted as a FASTA file. These sequences can be either contig or chromosome sequences. The FASTA format consists of two lines per record, the first being a sequence identifier and the second being the sequence itself. Ensure the sequence contains only valid nucleotide characters and no whitespace or newline characters.

4.2.3 Flat File

Annotated sequences can only be submitted in the EMBL flat file format. For the full range of features and qualifiers available for flat files and their expected content, please see WebFeat.

The feature table annotation must conform to the INSDC Feature Table Definition.

Some tools to help you create flat files are described in our Third Party Tools page.

4.2.4 Chromosome List File

The chromosome list file must be provided when the submission contains assembled chromosomes.

The file is a tab separated text file (USASCII7) up to four columns. An example chromosome list file, describing a eukaryote with four linear nuclear chromosomes and one linear mitochondrial chromosomes:

<table>
<thead>
<tr>
<th>OBJECT_NAME</th>
<th>CHROMOSOME_NAME</th>
<th>CHROMOSOME_TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr01</td>
<td>1 Linear-Chromosome</td>
<td></td>
</tr>
<tr>
<td>chr02</td>
<td>2 Linear-Chromosome</td>
<td></td>
</tr>
<tr>
<td>chr03</td>
<td>3 Linear-Chromosome</td>
<td></td>
</tr>
<tr>
<td>chr04</td>
<td>4 Linear-Chromosome</td>
<td></td>
</tr>
<tr>
<td>chrMi</td>
<td>MIT Linear-Chromosome Mitochondrion</td>
<td></td>
</tr>
</tbody>
</table>

Please read on for information on the content of the chromosome list file columns

- **OBJECT_NAME** (first column): The unique sequence name.
- **CHROMOSOME_NAME** (second column): The chromosome name. The value will appear as the /chromosome, /plasmid or /segment qualifier in the EMBL-Bank flat files. Names must:
  - match the pattern: ^[A-Za-z0-9][A-Za-z0-9_#-._]*$  
  - be shorter than 33 characters  
  - be unique within an assembly  
  - not contain any of the following as part of their name (case insensitive):
    - ‘chr’  
    - ‘chrm’  
    - ‘chrom’  
    - ‘chromosome’  
    - ‘linkage group’  
    - ‘linkage-group’  
    - ‘linkage_group’  
    - ‘plasmid’
- **CHROMOSOME_TYPE** (third column): Allowed values:
1. chromosome
2. plasmid
3. linkage_group
4. monopartite
5. segmented
6. multipartite

• **TOPOLOGY** (CHROMOSOME_TYPE modifier):
  - Topology is not a separate column but can be specified as a modifier to the chromosome type
  - Options are ‘linear’ or ‘circular’, default is linear
  - Must not conflict with any value specified in flat file
  - Contigs, scaffolds and transcriptome sequences are always linear: entering ‘circular’ here will be overriden

• **CHROMOSOME_LOCATION** (optional fourth column): By default eukaryotic chromosomes will be assumed to reside in the nucleus and prokaryotic chromosomes and plasmids in the cytoplasm. Allowed values:
  - Macronuclear
  - Nucleomorph
  - Mitochondrion
  - Kinetoplast
  - Chloroplast
  - Chromoplast
  - Plastid
  - Virion
  - Phage
  - Proviral
  - Prophage
  - Viroid
  - Cyanelle
  - Apicoplast
  - Leucoplast
  - Proplastid
  - Hydrogenosome
  - Chromatophore

### 4.2.5 AGP File

You may use an AGP file to describe the assembly of scaffolds from contigs, or of chromosomes from scaffolds. AGP files can be validated using the NCBI AGP validator.

The AGP file can also be used to define sequences as unplaced. Unplaced sequences are those known to be part of the assembly, but it is unknown which chromosome they belong to.
4.2.6 Unlocalised List File

This file should be provided when the submission contains chromosomes with unlocalised sequences. Unlocalised sequences are contigs or scaffolds that are associated with a specific chromosome but for which order and orientation is unknown. An example unlocalised list file:

<table>
<thead>
<tr>
<th>OBJECT_NAME</th>
<th>CHROMOSOME_NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>cb25.NA_084</td>
<td>III</td>
</tr>
<tr>
<td>cb25.NA_093</td>
<td>III</td>
</tr>
<tr>
<td>cb25.NA_108</td>
<td>III</td>
</tr>
</tbody>
</table>

The unlocalised list file is a tab separated text file (USASCII7) containing the following columns:

- **OBJECT_NAME** (first column): the unique sequence name matching a FASTA header or flatfile AC * line
- **CHROMOSOME_NAME** (second column): the unique chromosome name associated with this sequence. This must match with a CHROMOSOME_NAME in the chromosome list file.

4.2.7 Appendix: Unique Sequence Names

All sequences within one genome assembly submission must be identified by a unique sequence name provided in the FASTA, AGP or flat files.

It is essential that the sequence names are unique and used consistently between files. For example, the chromosome list file must refer to the chromosome sequences being submitted in FASTA, AGP or flat files using the unique entry name. Similarly, an AGP file must refer to scaffolds or contigs using unique entry names.

**FASTA**

The sequence name is extracted from the header line starting with >.

For example, the following sequence has name contig1:

```markdown
>contig1
AAACCCGGG...
```

**AGP**

The sequence name is extracted from the 1st (object) column.

**Flat Files**

The sequence name is extracted from the AC * line. The sequence name must be prefixed with a _ when using the flat file format.

For example, the following sequence has name contig1:

```
AC * _contig1
```

Note that for the AC * line, the ‘AC’ must be followed by exactly one space, an asterisk (*) character, and then one more space.
### 4.2.8 Appendix: Definition of Terms

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assembly</td>
<td>A set of chromosome assemblies, unlocalized and unplaced sequences, alternate loci and patches that represent a genome.</td>
</tr>
<tr>
<td>Assembly chain</td>
<td>The major and minor releases form an assembly chain. For example, the assembly accession for GRCh37 major release is GCA_000001405.1. The assembly accession consists of two parts: the assembly chain accession (GCA_000001405) and the assembly version (.1). The assembly version is incremented for each minor release while the assembly chain accession remains unchanged.</td>
</tr>
<tr>
<td>Chromosome</td>
<td>An assembled pseudomolecule that represents a biological chromosome. Most of the chromosome is expected to be represented by sequenced bases, although some gaps may still be present.</td>
</tr>
<tr>
<td>Placed sequence</td>
<td>A sequence that has a known chromosomal location and orientation.</td>
</tr>
<tr>
<td>Unplaced sequence</td>
<td>A sequence that is not associated with any specific chromosome.</td>
</tr>
<tr>
<td>Unlocalised sequence</td>
<td>A sequence that is associated with a specific chromosome without being ordered or oriented on that chromosome.</td>
</tr>
</tbody>
</table>

### 4.3 Flat File Example

The EMBL flat file format should be used for the submission of annotation data, from targeted assembled/annotated sequences to full annotated genome assemblies.
This page provides an example of the format, and may be used as a basis for preparing flat files for submission. When used for this purpose, please do not change any fields currently filled with ‘XXX’. These represent values which will be generated automatically by the ENA processing pipeline, so anything you enter here will be overwritten. On the other hand, be sure to edit any text enclosed in curly brackets (‘{’ and ‘}’).

To see the full range of available features and qualifiers, please visit WebFeat. To see any upcoming feature table changes see here.

For a list of third party tools which may help with the creation of your flat file or conversion from another format, please see our page on Third Party Tools.

We have some pre-prepared flat file templates available for use with common submission types available too, please see Entry Upload Templates.

<table>
<thead>
<tr>
<th>ID</th>
<th>XXX; XXX; {'linear' or 'circular'}; XXX; XXX; XXX; XXX.</th>
</tr>
</thead>
<tbody>
<tr>
<td>XX</td>
<td></td>
</tr>
<tr>
<td>AC</td>
<td>XXX;</td>
</tr>
<tr>
<td>XX</td>
<td></td>
</tr>
<tr>
<td>AC</td>
<td>* _{entry_name} (where entry_name=sequence name: e.g. _contig1 or _scaffold1)</td>
</tr>
<tr>
<td>XX</td>
<td></td>
</tr>
<tr>
<td>PR</td>
<td>Project:PRJEBNNNN;</td>
</tr>
<tr>
<td>XX</td>
<td></td>
</tr>
<tr>
<td>DE</td>
<td>XXX;</td>
</tr>
<tr>
<td>XX</td>
<td></td>
</tr>
<tr>
<td>RN</td>
<td>[1]</td>
</tr>
<tr>
<td>RP</td>
<td>1-2149</td>
</tr>
<tr>
<td>RA</td>
<td>XXX;</td>
</tr>
<tr>
<td>RT</td>
<td>;</td>
</tr>
<tr>
<td>RL</td>
<td>Submitted ((DD-MMM-YYYY)) to the INSDC.</td>
</tr>
<tr>
<td>XX</td>
<td></td>
</tr>
<tr>
<td>FH</td>
<td>Key Location/Qualifiers</td>
</tr>
<tr>
<td>FH</td>
<td></td>
</tr>
<tr>
<td>FT</td>
<td>source 1..588788</td>
</tr>
<tr>
<td>FT</td>
<td>/organism={&quot;scientific organism name&quot;}</td>
</tr>
<tr>
<td>FT</td>
<td>/mol_type=&quot;in vivo molecule type of sequence&quot;</td>
</tr>
<tr>
<td>XX</td>
<td></td>
</tr>
<tr>
<td>SQ</td>
<td>Sequence 588788 BP; 101836 A; 193561 C; 192752 G; 100639 T; 0 other;</td>
</tr>
<tr>
<td></td>
<td>tgcgtactcg aagagacgcg cccagattat ataagggcgt cgtctgagg cccagccgcg 60</td>
</tr>
<tr>
<td></td>
<td>gccggcgagt acgcgtgatc cacaaccqga acgcctggtc ggacgccgga ccgccgcggg 120</td>
</tr>
<tr>
<td></td>
<td>ggtttgctcgt gcctgcgcgg aagggcggg ccgcctgcgg gcgcctgcgg 180</td>
</tr>
<tr>
<td></td>
<td>[sequence truncated]...</td>
</tr>
</tbody>
</table>

### 4.4 Entry Upload Templates - Guidelines on Accepted Sequence Flat-file Formats

#### 4.4.1 Introduction

The EMBL flat file format is the only way to submit all types of functional sequence annotation to ENA. See the example flat file page for general information on this format.

The Entry Upload Templates provided on this page are skeletons of actual flatfiles, to adapt for submission. All submissions should include the topology of the ID line, the description from the DE line, and the feature table (FT lines). Other parts of the final flatfile will be collected during submission, including release date and references, rather than being present in the submitted file. This includes any fields filled with ‘XXX’: these can be left as they are and our pipeline will infer them later.
To use one of the templates, copy the boxed example and save into a plain text file. It is essential that you use a plain
text format, as hidden characters and formatting associated with other file types will result in automatic rejection of
your submission.

Please carefully read the accompanying notes for the template you choose, and make sure that all tokens (shown in
curly brackets {}) are replaced with actual values, and any unused parts of the FT block are removed. Please use
only the ASCII character set; extended ASCII (including special characters and characters not native to English) is not
accepted and may cause validation errors. Please use the nearest ASCII equivalent.

The templates provide the basic and minimal information required for constructing your flatfile. Advanced users may
wish to add further annotation within the scope of the feature table. In this case, please refer to the INS DC WebFeat
documentation.

Most templates here assume that the sequences are nuclear in origin. For organellar sequences, the qualifier/or-
ganelle="{organelle}" should be included, with the value of {organelle} taken from the INS DC controlled vocabulary.

Sequences extracted from mixed samples of anonymous organisms (metagenomic samples) where independent iden-
tification, such as culturing, has not been done are environmental samples by INS DC definition. The organism feature
for these sequences should reference a taxon whose name starts with ‘uncultured’. In addition, these should include
the flag ‘environmental sample’ within the source feature.

Sequences should always be submitted on the forward strand. When submitting a multi-feature region with loci on
both strands, you will need to apply the complement operator to the coordinates for reverse strand features. Refer to
the Feature Table Document or similarly-annotated records in the database.

The nucleotide sequence itself should comprise only characters from the IUPAC nucleotide nomenclature. An unse-
quenced region of unsure length should be represented with a string of 100 Ns; gaps of known length may use the
exact number, but if this is very large (e.g. >200bp) then it is advisable to condense the gap to one of unknown length
(100 Ns).

Unsequenced regions should be annotated with a gap feature, shown below. Gaps formed as artefacts of alignment
are not allowed. If the gap covers an entire intron, then the corresponding intron block should be removed from the
annotation.

<table>
<thead>
<tr>
<th>FT gap {X..Y}</th>
</tr>
</thead>
<tbody>
<tr>
<td>FT /estimated_length={unknown or known}</td>
</tr>
</tbody>
</table>

When you have prepared your flatfile, you will need to submit it using Webin-CLI.

Available templates:

- 1: HLA Gene With Sequenced Introns
- 2: HLA Gene With Unsequenced Introns
- 3: Bacterial Operon
- 4: Multi-Exon Gene
- 5: Multi-Feature Region
- 6: rDNA Region (ITS And rRNA Features)
- 7: Precursor mRNA
- 8: Synthetic Construct

**4.4.2 HLA Gene With Sequenced Introns**

HLA genes have strictly controlled formats in ENA flat files. Please adhere to the below conventions to avoid having
your submission rejected.
1. **{organism}** is a species-rank organism present in the database. Search available taxa at [ENA Search](https://www.ebi.ac.uk/ena/).  
2. **{gene}** has a value such as *HLA-A, HLA-B, HLA-DRB*  
3. **{allele}** takes the form `{gene}*{allele designation}`, e.g. *HLA-A*01:01, *HLA-B*40:32  
4. **{product}** takes the form *MHC class I antigen* or *MHC class II antigen*  
5. **{reading frame}** should be 1 where the start codon is present, otherwise 1, 2 or 3 depending on the reading frame of the 5’ partial coding sequence  
6. **{partial}** - delete if coding region is complete, otherwise just remove the curly brackets  
7. **{a to n}** are the exon/intron/CDS coordinates  
8. **{A and N}** are the coding region 5’ and 3’ coordinates, which may differ from exon coordinates due to UTR. If the start or stop codon is absent, please include partiality symbols, e.g. `<1..250` or `2917..>2960`  
9. Incompleteness of any features should be described using partiality symbols, e.g. `<1..250` or `2917..>2960`  
10. **{sequence}** should be the sequence in IUPAC nucleotide characters, as described above  
11. Add or remove exon and intron blocks as required

**HLA gene with sequenced introns:**

```plaintext
ID    XXX; XXX; linear; XXX; XXX; XXX; XXX.
XX
AC    XXX;
XX
DE    {organism} (partial) {gene} gene for {product}, allele {allele}
XX
FH    Key      Location/Qualifiers
FH    FT    source 1..{sequence length}
FT    /organism="{organism}"  
FT    /mol_type="genomic DNA"  
FT    CDS join(A..b,c..d,e..f,g..h,i..j,k..l,m..N)  
FT    /codon_start={reading frame}  
FT    /gene="{gene}"  
FT    /allele="{allele}"  
FT    /product="{product}"  
FT    /function="antigen presenting molecule"  
FT    exon {a..b}  
FT    /number=1  
FT    /gene="{gene}"  
FT    /allele="{allele}"  
FT    intron {b+1..c-1}  
FT    /number=1  
FT    /gene="{gene}"  
FT    /allele="{allele}"  
FT    exon {c..d}  
FT    /number=2  
FT    /gene="{gene}"  
FT    /allele="{allele}"  
FT    intron {d+1..e-1}  
FT    /number=2  
FT    /gene="{gene}"  
FT    /allele="{allele}"  
FT    exon {e..f}  
FT    /number=3  
FT    /gene="{gene}"  
```

(continues on next page)
4.4.3 HLA Gene With Unsequenced Introns

HLA genes have strictly controlled formats in ENA flat files. Please adhere to the below conventions to avoid having your submission rejected. When introns are unsequenced, exons should be concatenated in order with 100 Ns representing gaps of unknown length.

1. `{organism}` is a species-rank organism present in the database. Search available taxa at ENA Search.
2. `{gene}` has a value such as HLA-A, HLA-B, HLA-DRB
3. `{allele}` takes the form `{gene*allele designation}`, e.g. HLA-A*01:01, HLA-B*40:32
4. `{product}` takes the form MHC class I antigen or MHC class II antigen
5. `{reading frame}` should be 1 where the start codon is present, otherwise 1, 2 or 3 depending on the reading frame of the 5’ partial coding sequence
6. `{partial}` - delete if coding region is complete, otherwise just remove the curly brackets
7. `{a to h}` are the exon/intron/CDS coordinates
8. \{A and H\} are the coding region 5' and 3' coordinates, which may differ from exon coordinates due to UTR. If the start or stop codon is absent, please include partiality symbols, e.g. \(<1..250\) or \(2917..>2960\).

9. Incompleteness of any features should be described using partiality symbols, e.g. \(<1..250\) or \(2917..>2960\).

10. Add or remove exon and intron blocks as required.

11. \{sequence\} should be the sequence in IUPAC nucleotide characters, as described above.

---

```
ID XXX; XXX; linear; XXX; XXX; XXX; XXX.
XX
AC XXX;
XX
DE {organism} (partial) {gene} gene for {product}, allele {allele}
XX
FH Key Location/Qualifiers
FH
FT source 1..{sequence length}
FT /organism="{organism}"
FT /mol_type="genomic DNA"
FT CDS join(A..b,b+101..c,c+101..d,d+101..e,e+101..f,f+101..g,g+101..H)
FT /codon_start={reading frame}
FT /gene="{gene}"
FT /allele="{allele}"
FT /product="{product}"
FT /function="antigen presenting molecule"
FT exon {a..b}
FT /number=1
FT /gene="{gene}"
FT /allele="{allele}"
FT gap b+1..b+100
FT /estimated_length=unknown
FT exon {b+101..c}
FT /number=2
FT /gene="{gene}"
FT /allele="{allele}"
FT gap c+1..c+100
FT /estimated_length=unknown
FT exon {c+101..d}
FT /number=3
FT /gene="{gene}"
FT /allele="{allele}"
FT gap d+1..d+100
FT /estimated_length=unknown
FT exon {d+101..e}
FT /number=4
FT /gene="{gene}"
FT /allele="{allele}"
FT gap e+1..e+100
FT /estimated_length=unknown
FT exon {e+101..f}
FT /number=5
FT /gene="{gene}"
FT /allele="{allele}"
FT gap f+1..f+100
FT /estimated_length=unknown
FT exon {f+101..g}
FT /number=6
FT /gene="{gene}"
```

(continues on next page)
4.4.4 Bacterial Operon

Bacterial operons contain multiple ORFs which are transcribed together. In ENA records, the annotation contains an operon feature and multiple CDS features.

1. {organism} is a species-rank organism present in the database. Search available taxa at ENA Search.
2. {strain} is the strain identifier. You may change this to /isolate if you prefer.
3. {operon} is the name of the operon and has a value like gal or lac.
4. {geneX} is the gene symbol and has a value like galA or lacZ.
5. {productX} is the product/protein name and has a value like galactose-permease, beta-galactosidase.
6. {reading frame} should be 1 where the start codon is present, otherwise 1, 2 or 3 depending on the reading frame of the 5’ partial coding sequence.
7. {a to l} are the coordinates of the operon and CDS 5’ and 3’ ends.
8. Incompleteness of any features should be described using partiality symbols, e.g. <1..250 or 2917..>2960.
9. {sequence} should be the sequence in IUPAC nucleotide characters, as described above.
10. Add or remove CDS blocks as required.
11. Advanced users: consider including other features, such as promoter, -35_signal and -10_signal. See the WebFeat page for more information.

Optional features

| FT   | /allele="\{allele\}" |
| FT   | gap a..l |
| FT   | /estimated_length=unknown |
| FT   | /number=7 |
| FT   | /gene="\{gene\}" |
| XX   | SQ |

{sequence}
### 4.4.5 Multi-Exon Gene

This template should be used as a starting point for submitting multi-exon genes. Where optional qualifiers are not used, please remove the entire line from the template.

1. *{organism}* is a species-rank organism present in the database. Search available taxa at ENA Search.

2. *{identifier}* is a form of sample or organism identification. You must use at least one from the below list and fill in the *(ID)* field with it. Other qualifiers are available in WebFeat.
   a. Isolate: any sample or isolate name
b. Strain: the strain of the sequenced organism

c. Clone: the clone name of the sequence

d. Note: the breed of a domesticate

e. Cultivar: a cultivated variety of a plant or fungus

f. Variety: a natural variety of a plant or fungus

3. \{gene\} is the gene symbol, putative or official. This is optional, but highly recommended

4. \{product\} is the product/protein name and is mandatory. If uncertain, use “hypothetical protein”

5. \{allele\} is the allele name. This is optional.

6. \{reading frame\} should be 1 where the start codon is present, otherwise 1, 2 or 3 depending on the reading frame of the 5’ partial coding sequence

7. \{partial\} - delete if coding region is complete, otherwise just remove the curly brackets

8. \{a to n\} are the exon/intron/CDS coordinates

9. \{A to N\} are the coding region 5’ and 3’ coordinates, which may differ from exon coordinates due to UTR. If the start or stop codon is absent, please include partiality symbols, e.g. \(<1..250 \text{ or } 2917..>2960\)

10. Incompleteness of any features should be described using partiality symbols, e.g. \(<1..250 \text{ or } 2917..>2960\)

11. \{sequence\} should be the sequence in IUPAC nucleotide characters, as described above

12. Add or remove exon, intron and gap blocks as required

Multi-exon genes

<table>
<thead>
<tr>
<th>ID</th>
<th>XXX; XXX; linear; XXX; XXX; XXX; XXX.</th>
</tr>
</thead>
<tbody>
<tr>
<td>XX</td>
<td></td>
</tr>
<tr>
<td>AC</td>
<td>XXX;</td>
</tr>
<tr>
<td>XX</td>
<td></td>
</tr>
<tr>
<td>DE</td>
<td>{organism} {partial} {gene} gene \textbf{for} {product}, {identifier} {ID}</td>
</tr>
<tr>
<td>XX</td>
<td></td>
</tr>
<tr>
<td>FH</td>
<td>Key Location/Qualifiers</td>
</tr>
<tr>
<td>FH</td>
<td></td>
</tr>
<tr>
<td>FT</td>
<td>source {sequence length}</td>
</tr>
<tr>
<td>FT</td>
<td>/organism=&quot;{organism}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/mol_type=&quot;genomic DNA&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/{identifier}=&quot;{ID}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>CDS join(A..b,c..d,e..f,g..h,i..j,k..l,m..N)</td>
</tr>
<tr>
<td>FT</td>
<td>/codon_start={reading frame}</td>
</tr>
<tr>
<td>FT</td>
<td>/gene=&quot;{gene}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/product=&quot;{product}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/allele=&quot;{allele}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/function=&quot;{function}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>exon {a..b}</td>
</tr>
<tr>
<td>FT</td>
<td>/number=1</td>
</tr>
<tr>
<td>FT</td>
<td>/gene=&quot;{gene}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/allele=&quot;{allele}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>intron {b+1..c-1}</td>
</tr>
<tr>
<td>FT</td>
<td>/number=1</td>
</tr>
<tr>
<td>FT</td>
<td>/gene=&quot;{gene}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/allele=&quot;{allele}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>exon {c..d}</td>
</tr>
<tr>
<td>FT</td>
<td>/number=2</td>
</tr>
<tr>
<td>FT</td>
<td>/gene=&quot;{gene}&quot;</td>
</tr>
</tbody>
</table>

(continues on next page)
4.4.6 Multi-Feature Region

1. `{organism}` is a species-rank organism present in the database. Search available taxa at ENA Search.

2. `{organelle}` with the value taken from the INSDC controlled vocabulary. The organelle should also be added to the DE line. Remove this entirely if the sequence is of nuclear origin.

3. `{identifier}` is a form of sample or organism identification. You must use at least one from the below list and fill in the `{ID}` field with it. Other qualifiers are available in WebFeat.
   a. Isolate: any sample or isolate name
b. Strain: the strain of the sequenced organism

c. Clone: the clone name of the sequence

d. Note: the breed of a domesticate

e. Cultivar: a cultivated variety of a plant or fungus

f. Variety: a natural variety of a plant or fungus

4. **{gene}** is the gene symbol, putative or official. It is optional, but highly recommended. For tRNAs, the INSDC standard is \textit{tRNA-Aaa} where \textit{Aaa} is the 3-letter amino acid code (e.g. \textit{tRNA-Gly}). For rRNAs, the standard is \textit{XXS ribosomal RNA}, where \textit{XX} is the sedimentation coefficient (e.g. \textit{16S ribosomal RNA})

5. **{product}** is the product/protein name and is mandatory. If uncertain, use “hypothetical protein”

6. **{reading frame}** should be 1 where the start codon is present, otherwise 1, 2 or 3 depending on the reading frame of the 5’ partial coding sequence

7. **{a to h}** are the feature coordinates. For CDS features, this is the first and last base of the coding sequence, whether or not the start or stop codons are present.

8. **{short note}** should describe any misc_feature in simple terms, e.g. intergenic spacer region. It is useful here to refer to similar entries in the database. For tRNAs, the INSDC standard is \textit{tRNA-Aaa} where \textit{Aaa} is the 3-letter amino acid code (e.g. \textit{tRNA-Gly}). For rRNAs, the standard is \textit{XXS ribosomal RNA}, where \textit{XX} is the sedimentation coefficient (e.g. \textit{16S ribosomal RNA})

9. Incompleteness of any features should be described using partiality symbols, e.g. <1..250 or \textgreater{}2917..>2960

10. **{sequence}** should be the sequence in IUPAC nucleotide characters, as described above

11. Any of the feature blocks can be replicated/removed as required

12. The DE line should be written using the templated format but will need to be expanded according to the number and type of features in the sequence

Multi-feature region:

<table>
<thead>
<tr>
<th>ID</th>
<th>XXX; XXX; linear; XXX; XXX; XXX; XXX.</th>
</tr>
</thead>
<tbody>
<tr>
<td>XX</td>
<td>AC XXX; X</td>
</tr>
<tr>
<td>DE</td>
<td>{organism} {organelle} {partial} {gene1} gene, {gene2} gene, {gene3} gene and \rightarrow {short note}, {identifier} {ID}</td>
</tr>
<tr>
<td>XX</td>
<td>FH Key Location/Qualifiers</td>
</tr>
<tr>
<td>FH</td>
<td>FT source 1..{sequence length}</td>
</tr>
<tr>
<td>FT</td>
<td>/organism=&quot;{organism}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/mol_type=&quot;genomic DNA&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/{identifier}=&quot;{ID}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>CDS a..b</td>
</tr>
<tr>
<td>FT</td>
<td>/codon_start={reading frame}</td>
</tr>
<tr>
<td>FT</td>
<td>/gene=&quot;{gene1}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/product=&quot;{product1}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>tRNA c..d</td>
</tr>
<tr>
<td>FT</td>
<td>/gene=&quot;{gene2}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/product=&quot;{product2}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>rRNA e..f</td>
</tr>
<tr>
<td>FT</td>
<td>/gene=&quot;{gene3}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/product=&quot;{product3}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>misc_feature g..h</td>
</tr>
</tbody>
</table>

(continues on next page)
4.4.7 rDNA Region (ITS And rRNA Features)

The ITS/rDNA region can be submitted using an annotation checklist but this provides all details within a single misc_RNA feature. If you wish to add individual feature annotation for each rRNA and ITS, you will need to generate the file yourself. This section provides a skeleton for that.

1. {organism} is a species-rank organism present in the database. Search available taxa at ENA Search.

2. {identifier} is a form of sample or organism identification. You must use at least one from the below list and fill in the {ID} field with it. Other qualifiers are available in WebFeat.
   a. Isolate: any sample or isolate name
   b. Strain: the strain of the sequenced organism
   c. Clone: the clone name of the sequence
   d. Note: the breed of a domesticate
   e. Cultivar: a cultivated variety of a plant or fungus
   f. Variety: a natural variety of a plant or fungus

3. {a to f} are the feature coordinates. Incompleteness of any features (rRNA, misc_RNA) should be described using partiality symbols, e.g. <1..250 or 2917..>2960

4. {sequence} should be the sequence in IUPAC nucleotide characters, as described above

5. Any of the feature blocks can be replicated/removed as required

6. Please update the DE line to reflect the addition/removal of features

rDNA region:

<table>
<thead>
<tr>
<th>ID</th>
<th>XXX; XXX; linear; XXX; XXX; XXX; XXX.</th>
</tr>
</thead>
<tbody>
<tr>
<td>XX</td>
<td>XXX;</td>
</tr>
<tr>
<td>AC</td>
<td>XXX;</td>
</tr>
<tr>
<td>XX</td>
<td></td>
</tr>
<tr>
<td>DE</td>
<td>{organism} 18S rRNA gene, ITS1, 5.8S rRNA gene, ITS2 and 28S rRNA gene,</td>
</tr>
<tr>
<td></td>
<td>{identifier} {ID}</td>
</tr>
<tr>
<td>XX</td>
<td></td>
</tr>
<tr>
<td>FH</td>
<td>Key Location/Qualifiers</td>
</tr>
<tr>
<td>FT</td>
<td>source 1..{sequence length}</td>
</tr>
<tr>
<td>FT</td>
<td>/organism=&quot;{organism}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/mol_type=&quot;genomic DNA&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/{identifier}=&quot;{ID}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>rRNA a..b</td>
</tr>
<tr>
<td>FT</td>
<td>/gene=&quot;18S rRNA&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/product=&quot;18S ribosomal RNA&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>misc_RNA b+1..c</td>
</tr>
<tr>
<td>FT</td>
<td>/note=&quot;internal transcribed spacer 1, ITS1&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>rRNA c+1..d</td>
</tr>
<tr>
<td>FT</td>
<td>/gene=&quot;5.8S rRNA&quot;</td>
</tr>
</tbody>
</table>
4.4.8 Precursor mRNA

Precursor mRNA can be submitted using an annotation checklist called “Single-CDS mRNA” but this does not annotation of features that arise from processing of the transcript, such as sig_peptide or mat_peptide features. To add these, you will need to prepare the file yourself with the template below.

1. \{organism\} is a species-rank organism present in the database. Search available taxa at ENA Search.
2. \{organelle\} with the value taken from the INSDC controlled vocabulary. The organelle should also be added to the DE line. Remove this entirely if the sequence is of nuclear origin.
3. \{identifier\} is a form of sample or organism identification. You must use at least one from the below list and fill in the \{ID\} field with it. Other qualifiers are available in WebFeat.
   a. Isolate: any sample or isolate name
   b. Strain: the strain of the sequenced organism
   c. Clone: the clone name of the sequence
   d. Note: the breed of a domesticate
   e. Cultivar: a cultivated variety of a plant or fungus
   f. Variety: a natural variety of a plant or fungus
   g. Dev_stage: the developmental stage of the organism
   h. Tissue_type: the tissue type sampled
   i. Cell_type: the type sampled
   j. Sex: the sex of the animal
   k. Mating_type: the mating type of the prokaryote/lower eukaryote
4. \{a to l\} are the feature coordinates. Incompleteness of any features (CDS, sig_peptide, mat_peptide) should be described using partiality symbols, e.g. <1..250 or 2917..>2960
5. \{gene\} is the gene symbol, putative or official. It is optional, but highly recommended. If you remove this line, also remove it from the DE line.
6. \{precursor\} is the protein precursor name, and is mandatory
7. \{reading frame\} should be 1 where the start codon is present, otherwise 1, 2 or 3 depending on the reading frame of the 5’ partial coding sequence
8. \{mat_peptide N\} is the name of the Nth mature peptide
9. \{partial\} - delete if coding region is complete, otherwise just remove the curly brackets
10. \{sequence\} should be the sequence in IUPAC nucleotide characters, as described above
Precursor mRNA:

<table>
<thead>
<tr>
<th>ID</th>
<th>XXX; XXX; linear; XXX; XXX; XXX; XXX.</th>
</tr>
</thead>
<tbody>
<tr>
<td>XX</td>
<td>XXX;</td>
</tr>
<tr>
<td>AC</td>
<td>XXX;</td>
</tr>
<tr>
<td>XX</td>
<td></td>
</tr>
<tr>
<td>DE</td>
<td>{organism} {partial} mRNA for {precursor} {gene} gene, {identifier} {ID}</td>
</tr>
<tr>
<td>FH</td>
<td>Key Location/Qualifiers</td>
</tr>
<tr>
<td>FT</td>
<td>source 1..{sequence length}</td>
</tr>
<tr>
<td>FT</td>
<td>/organism=&quot;{organism}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/organelle=&quot;{organelle}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/mol_type=&quot;genomic DNA&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/{identifier}=&quot;{ID}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>CDS a..b</td>
</tr>
<tr>
<td>FT</td>
<td>/codon_start={reading frame}</td>
</tr>
<tr>
<td>FT</td>
<td>/gene=&quot;{gene}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/product=&quot;{precursor}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>sig_peptide e..f</td>
</tr>
<tr>
<td>FT</td>
<td>/gene=&quot;{gene}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>mat_peptide g..h</td>
</tr>
<tr>
<td>FT</td>
<td>/gene=&quot;{gene}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/product=&quot;{mat_peptide 1}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>mat_peptide i..j</td>
</tr>
<tr>
<td>FT</td>
<td>/gene=&quot;{gene}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/product=&quot;{mat_peptide 2}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>mat_peptide k..l</td>
</tr>
<tr>
<td>FT</td>
<td>/gene=&quot;{gene}&quot;</td>
</tr>
<tr>
<td>FT</td>
<td>/product=&quot;{mat_peptide 3}&quot;</td>
</tr>
<tr>
<td>SQ</td>
<td>{sequence}</td>
</tr>
</tbody>
</table>

### 4.4.9 Synthetic Construct

You should use this template if your sequence has been constructed synthetically and includes artificial genes and vectors. The submitted sequence must have been validated by nucleotide sequencing for acceptance.

This annotation is made up of one or more source features, and generally includes coding regions and various components described with misc_features. If more than one source feature is used, the first should describe the molecule as a whole and be labelled with the /focus qualifier. The molecule type (/mol_type) for synthetic constructs is either "other DNA" or "other RNA".

The example below is for a gene which has been edited for preferential expression in another organism. The first source describes the synthetic nature, the second describes the origin of the sequence (of which there can be many if parts are constructed from different organisms.

1. **{other DNA or RNA}** refers to the molecule type, and also appears in the ID line. Use *other DNA* or *other RNA* as the value as appropriate.

2. Note that the /focus source has organism name *synthetic construct*. This is the standard taxon for this type of sequence. If you have named a vector sequence specifically, e.g. *Cloning vector ABC*, you may use this in place and a taxon will be added to the database according to NCBI Taxonomy’s operating procedures.
3. {organism} is a species-rank organism present in the database referring to the originating organism. Search available taxa at ENA Search.

4. {a to h} are the feature coordinates. Incompleteness of any features (except source) should be described using partiality symbols, e.g. <1..250 or 2917..>2960

5. {gene} is the gene symbol, putative or official. It is optional, but highly recommended. If you remove this line, also remove it from the DE line.

6. {product} is the protein/product name. This is mandatory for CDS.

7. {table} is the translation table under which the coding region is translated. Learn more at the Translation Tables page

8. {short note 1} should provide additional information on the promoter, e.g. Eu and SRalpha promoter

9. {short note 2} should provide additional information for the CDS, e.g. preferential codon usage changed for expression in {organism 2}

10. {short note 3} should support the miscellaneous feature, e.g. additional stop codon

11. {sequence} should be the sequence in IUPAC nucleotide characters, as described above

Synthetic construct:

```
ID XXX; XXX; linear; XXX; XXX; XXX; XXX. 
XX AC XXX; 
XX DE Synthetic construct for {organism} {gene} gene for {product} 
XX FH Key Location/Qualifiers 
FH FT source 1..{sequence length} 
FT /organism="synthetic construct" 
FT /mol_type="{other DNA or RNA}" 
FT /focus 
FT source a..b 
FT /organism="{organism}" 
FT /mol_type="other DNA" 
FT /note="{short note 1}" 
FT CDS e..f 
FT /transl_table={table} 
FT /gene="{gene}" 
FT /product="{product}" 
FT /note="{short note 2}" 
FT misc_feature g..h 
FT /note="{short note 3}"
XX SQ {sequence} //
```

### 4.5 Preparing A File For Upload

For submissions using the interactive or RESTful submission services, it is necessary to transfer data files to ENA, after which you can formally submit them. Guidance on how to perform the upload step is given below.
Please note that if you are submitting via Webin-CLI, you do not need to perform this upload step as it is handled for you by the program. However, Webin-CLI will still require you to compress some files, which you can learn how to do below.

Most files uploaded to the ENA FTP server need to:

1. Be compressed
2. Have their MD5 checksum registered

### 4.5.1 Step 1: Compress The File Using gzip Or bzip2

Files that are in a human-readable text format (such as FASTQ or FASTA) must be compressed before they are uploaded to the ENA FTP server. Files that are not in a human-readable text format like BAM, CRAM, SFF are already in a compact format so additional compression should not be used. With the exception of Oxford Nanopore FAST5 files, do not tar archive any collections of files - each should be uploaded separately.

If you are unsure about the format that your files should be in, please view our read data formats page. This includes advice on general read data formats, as well as information specific to platforms or sequencing applications.

Tools for compressing files in gzip or bzip2 format are included in Linux and Mac distributions. Below is a simple example of the appropriate syntax for compressing a file named `eg_01.fq`:

```bash
gzip eg_01.fq
```

The result of this would be a compressed file named `eg_01.fq.gz`. Files can always be decompressed later:

```bash
gunzip eg_01.fq.gz
```

The `--k` flag can be used to retain the original file. This results in two files: the original uncompressed file and a compressed copy:

```bash
gzip --k eg_01.fq
```

For Windows users, third party tools are available to accomplish this. The standard compression type used in Windows, ZIP, is not accepted in ENA submissions.

### 4.5.2 Step 2: Record The MD5 Checksum For The File

MD5 is a function which can be applied to a file to create a 32 character string. This string is unique to the file and functions like a fingerprint: if the contents of the file change in any way the MD5 checksum will change as well. The file name can change without affecting the MD5 value because the calculation is done on the contents of the file only. You must provide the MD5 value for your files so that we can be sure they transferred successfully and completely.

Calculating and registering an MD5 value is not necessary if you are using Webin-CLI, as the program handles this for you behind the scenes.

Functions for calculating MD5 value are included with Linux and Mac distributions. Below you can see an example command, along with its output:

```bash
md5 eg_01.fq.gz
MD5 (eg_01.fq.gz) = 74f085a6f3dd8b2877b89fcb592c7f5c
```

Note that the MD5 value for this file is `74f085a6f3dd8b2877b89fcb592c7f5c`.

The command in the below box generates no output, because the user directs the output to a file with the same name as the compressed FASTQ file but with `.md5` appended. Creating and uploading a `.md5` file is one way you can register your file’s MD5 value.
Sometimes the correct command is `md5sum` instead of `md5`:

```
md5sum eg_01.fq.gz
MD5 (eg_01.fq.gz) = 74f085a6f3dd8b2877b89fcb592c7f5c
```

For Windows users, 3rd party tools can be found to carry out these checks.

### Registering The MD5 Checksum With ENA

As described above, you can register your file’s MD5 value by outputting it to a second file and uploading this along with the data file. Alternatively, you can make a note of the value and enter it when prompted during the submission process.

The Webin File Uploader is a program provided by ENA covered here which can be used to upload your files. It also automatically creates `.md5` files and uploads them for you.

If you make and upload your own `.md5` file, be sure it contains only the 32 digit MD5 value for a single file, and that its name matches the name of that file.

### 4.5.3 File Validation Errors

Improperly compressed files and incorrect MD5 values are two of the most common errors encountered when processing submitted read files. Often, this happens because the file upload was incomplete. When this occurs, you are automatically notified by email. Guidance on fixing such errors is given in our common run submission errors page.

### 4.5.4 Next Steps

Once you have compressed your files and recorded their MD5 values in some way, proceed to upload them:

Data Upload Documentation

### 4.5.5 Fair Use Policy

ENA is a permanent and comprehensive data repository for public domain sequence and associated information. Data submitted into this system are routed transiently through users’ private data upload areas until such time as a submission instruction from the user, through his/her use of the Webin system, leads to the validation of the data and the movement of files from the data upload area into the ENA.

The data upload areas are provided as a temporary place in which data are held while in transit. As such, they are neither intended nor suitable for any longer-term storage of data. Such storage is provided in ENA itself. Once in ENA, data can be released immediately following submission or can be held confidential prior to analysis and literature publication if required.

We expect any given data file to remain in a data upload area for no longer than 2 months before the instruction is given by the user to submit the file. While we attempt to remind users of this policy at the 2 months time point we reserve the right to routinely delete any data files that persist in them for more than 2 months.

We place no absolute limit within the 2-month period on the total volume of user data that may exist in a data upload area at any one time and are keen to accommodate the largest submissions where possible. However, we strongly encourage continuous data submissions where files are uploaded and submitted in small patches of few Terabytes or less and expect that volumes would not exceed 10 Terabytes under normal circumstances. Unexpected heavy use of
the data upload areas may have an impact on other Webin users and we are grateful for users’ attention to this aspect of our fair use policy.

4.6 Uploading Files To ENA

4.6.1 Introduction

You must upload data files into your private Webin file upload area at EMBL-EBI before you can submit the files through the Webin submission service. The most user-friendly approach is Using Webin File Uploader.

Please note that this is not necessary if you are using Webin-CLI, as it handles the upload process for you.

All upload methods described below will require you to have registered a Webin account, the ID for which resembles ‘Webin-XXXXXX’. Always enter this with an upper-case ‘W’. Register an account or reset your password at the Webin login page.

Keep Local Copies

Always keep a local copy of the uploaded files until the files have been successfully submitted and archived. The Webin file upload area is a temporary transit area which is not backed up. Any files on the area are subject to our fair use policy (see below).

4.6.2 File Upload Options

You will upload files to your private Webin file upload area using either FTP or Aspera protocol through the webin.ebi.ac.uk service. The authentication is done using your Webin submission account name and password.

Your username resembles ‘Webin-XXXXXX’. Register an account or reset your password at the Webin login page.

There are a number of ways to accomplish the upload, detailed below:

- Using Webin File Uploader
- Uploading Files Using Command Line FTP Client
- Using FileZilla On Windows
- Using Aspera ascp Command Line Program
- Using Windows File Explorer

If you have problems with using these services, you may find help in the appendix:

- Appendix: Configuring Your Firewall For ENA Upload

Using Webin File Uploader

Files can be uploaded using the Webin File Uploader, a Java web start application downloadable from Interactive Webin or the below link. To use this program, follow the below instructions in video or text form:

1. Download the Webin File Uploader
2. Launch the application (Mac users please see further instructions below),
3. Enter your Webin username in the Username field, ensuring the ‘W’ is upper-case.
4. Enter your Webin password in the Password field.
5. Browse into the local Upload Directory containing the data files you wish to upload using the ... button.
6. Click ‘okay’ to see the list of all the files contained in the selected directory displayed in the Webin File Uploader window
7. Choose Overwrite option if you wish to replace any existing files which have been previously uploaded.
8. Choose Upload Tree option if you wish to preserve the directory structure when uploading files to the Webin upload area. By default, the files will be uploaded into the root directory of your Webin upload area.
9. Select the files to upload. You can use the Select All button to select all the files for upload.
10. Click on the Upload button.

Additional Instructions For Mac Users

When downloading the application the following dialog box will be displayed:

Select the Save File option to save the WebinUploader.jnlp file to your local download directory.
If you selected the Open with option instead of the Save File option then the following dialog box will be displayed:
In this case please select OK. This will save the WebinUploader.jnlp file to your default local download directory.

In order to run the File Uploader application, open your file explorer and go to the directory where the WebinUploader.jnlp file has been saved.

While pressing the ctrl button, select the WebinUploader.jnlp file then select the open option.

The following dialog will now be displayed:

Now select the Open button. This will launch the the Webin File Uploader application.

### Uploading Files Using Command Line FTP Client

This section explains how to upload files to us using a command line FTP client in Linux or Mac. The built in FTP tool for Windows command line does not support FTPS so Windows users are recommended to use an alternative:

- **Using Webin File Uploader**
- **Using FileZilla On Windows**

The below instructions describe how you may upload your files to us through a command line FTP client in Linux or Mac.

1. Open a terminal and type `lftp webin2.ebi.ac.uk -u Webin-xxxxx`, filling in your Webin username
2. Enter your password when prompted
3. Type `ls` command to check the content of your drop box.
4. Use `mput <filename>` command to upload files.
5. Use ```bye``` command to exit the ftp client.

Note that in your Webin username, the ‘W’ should be upper case.

**Using FileZilla On Windows**

FileZilla allows you to transfer files via FTP through a user-friendly graphical interface.

1. Download and install FileZilla. If you are not administrator of your computer then download the portable version of FileZilla.

2. Use the binary mode: Transfer menu -> Transfer Type -> Binary.

3. Open the ‘Site Manager’ menu with the button at the top-left

4. Use the ‘New Site’ option

5. Enter details to match those in the screenshot, adding your own Webin username and password:

6. Click ‘Connect’

7. Search for the file(s) you want to upload using the tree on the left panel.

8. Create directories in your drop box (if necessary) using the tree on the right panel.

9. Drag and drop the files you want to upload from the lower left panel to the lower right panel.

10. Once your transfer is successful, close the application.

**Using Aspera ascp Command Line Program**

Aspera is a commercial file transfer protocol that may provide better transfer speeds than FTP.

---

4.6. Uploading Files To ENA
Download Aspera CLI from [here](#).

Please select the correct operating system. The `ascp` command line client is distributed as part of the Aspera Cli in the `cli/bin` folder.

Your command should look similar to this:

```
ascp -QT -l300M -L- <file(s)> <Webin-N>@webin.ebi.ac.uk:.
```

The `-l300M` option sets the upload speed limit to 300MB/s. You may wish to lower this value to increase the reliability of the transfer.

The `-L-` option is for printing logs out while transferring.

The `<file(s)>` can be a file mask (e.g. `*.cram`), a list of files or a single file.

`<Webin-N>` is your Webin submission account name.

### Using Windows File Explorer

Right click [Computer](#) and select [Add a network location](#) from the menu.

Click [Next](#).
Select **Choose a custom network location** and click **Next**

Type `ftp://webin.ebi.ac.uk` in the **Internet or network address** field and click **Next**
Unselect Log on anonymously, type your Webin user name in the User name field and click Next.

Type a network location to show in Windows Explorer e.g. webin.ebi.ac.uk then click Next.
Click Finish

When using the new folder you will prompted for your Webin password. Type your password and click Log on
Appendix: Configuring Your Firewall For ENA Upload

While most users should not encounter problems in this area, it may sometimes be necessary to configure your firewall to permit upload of data to ENA. Users attempting to connect from an institutional network may find that their IT services department has placed restrictions on their ability to connect to FTP services. This information could be useful in getting our service whitelisted.

FTP is used in passive mode and connection will be opened to one of the below ports:

- 40000
- 50000

Access to port 21 is required for the following IP address (webin2.ebi.ac.uk):

- 193.62.193.143

4.6.3 File MD5 Checksums Value

Large file transfers do not always complete successfully over the internet.

An MD5 checksum can be computed for a file before and after transfer to verify that the file was transmitted successfully. You must provide an MD5 value for each file submitted to the archive. We will re-compute and verify the MD5 checksum to make sure that the file transfer was completed without any changes to the file contents.

Calculating this value is covered in Preparing A File For Upload.

4.6.4 Invalid File Submissions

All files are subjected to a format check done by Webin. This does not occur until after you formally submit the files. You will generally be notified of this by email, and may be required to correct and reupload the file. Advice on this can be found in our Common Run Submission Errors FAQ

4.6.5 Fair Use Policy

ENA is a permanent and comprehensive data repository for public domain sequence and associated information. Data submitted into this system are routed transiently through users’ private data upload areas until such time as a
submission instruction from the user, through his/her use of the Webin system, leads to the validation of the data and the movement of files from the data upload area into the ENA.

The data upload areas are provided as a temporary place in which data are held while in transit. As such, they are neither intended nor suitable for any longer-term storage of data. Such storage is provided in ENA itself. Once in ENA, data can be released immediately following submission or can be held confidential prior to analysis and literature publication if required.

We expect any given data file to remain in a data upload area for no longer than 2 months before the instruction is given by the user to submit the file. While we attempt to remind users of this policy at the 2 months time point we reserve the right to routinely delete any data files that persist in them for more than 2 months.

We place no absolute limit within the 2-month period on the total volume of user data that may exist in a data upload area at any one time and are keen to accommodate the largest submissions where possible. However, we strongly encourage continuous data submissions where files are uploaded and submitted in small patches of few Terabytes or less and expect that volumes would not exceed 10 Terabytes under normal circumstances. Unexpected heavy use of the data upload areas may have an impact on other Webin users and we are grateful for users’ attention to this aspect of our fair use policy.
5.1 Introduction

To submit raw read sequencing data to ENA you must also provide some metadata to describe your sequencing project. This helps make your data re-useable and searchable.

Within ENA, raw reads are represented as ‘run’ and ‘experiment’ submission objects. The run submission holds information about the raw read files generated in a run of sequencing.
The **experiment** submission holds metadata that describe the methods used to sequence the sample.

If you are not yet familiar with the metadata model, please see [here](#) for some more information.

As a raw read submission references ENA sample and study objects, you must submit these before your submit your read data. See below for information on how to register a study within ENA to describe your overall research project and samples with information on the biological data that was sequenced:

- Register a study
- Register a sample

## 5.2 Accessions

Once the raw reads are registered, Webin will report two unique accession numbers for each read submission. The first starts with ERR and is called the Run accession. The other starts with ERX and is called the Experiment accession.

Always make a note of any accessions you receive as these are the unique identifiers for each of your submissions to ENA.

## 5.3 Submission Options

Reads can be submitted in one of three ways. For an overview of these, please see the [General Guide on Submitting to ENA](#).

### 5.3.1 Submit Raw Reads Interactively

Read files are represented in the database as run objects which point to the file location in an FTP directory. Runs also point to experiments, which describe the library preparation protocol and themselves point to the sample and study for the data, as described in the below image:
Before you register the run and experiment objects, you should have completed the following steps:

- Register a study
- Register a sample
- Upload your read files

To start your read submission, log in to your Webin account and select the ‘Submit sequence reads and experiments’ radial in the ‘New Submission’ tab.
Step 1: Choose A Study To Add To

1. Select the study you wish to submit your data to. You may only add data to one study in each submission.

2. Click next to proceed to the next step, where you can register samples. It is recommended that you do this in advance, for which guidance can be found in the section Sample Registration. If you have already registered your samples, click ‘Skip’.

Step 2: Describe Your Experiments

In this step, you will register the files you uploaded so that they may be archived, and provide metadata explaining how they were produced.

1. First, you will need to select the file format your runs are to be submitted in:

2. If you switch to any of the other tabs while filling out run information, the information you have already entered in the New Submission tab will be preserved:
3. You can download a template spreadsheet to enter details of your submission into, then reupload it later to complete your submission.

4. The third button will download the spreadsheet containing any information you have already entered.

5. Fill out each field to provide a description of your experiment.

6. The ‘Sample Reference’ box must contain the accession or alias of a sample you have registered with us.

7. The ‘Sample Reference Suggestions’ box will contain references to any samples you registered in the previous step, but will otherwise be empty. In the latter case, look up the appropriate aliases and enter them manually.

8. Use the green cross to add more rows.

9. The file name must exactly correspond with the name of a file in your upload directory including any subdirectories (see the top of this page if you have not uploaded your files).

10. The checksum is a ‘fingerprint’ of the file which will allow us to verify that it was uploaded successfully. In Linux and Mac, you can generate this value from the command line by running the command ‘md5’ or ‘md5sum’ on the file, while Microsoft has a support article on performing this activity for Windows.

11. When you have filled out all the information you need to, click ‘Submit’ to finish submission. Accession numbers will be provided as soon as the submission is completed.

### 5.3.2 Submit Raw Reads with Webin-CLI

#### Introduction

Sequence read data can be submitted to the European Nucleotide Archive (ENA) using the Webin command line submission interface with `--context reads` option.

A sequence read submission consists of:

- General experiment information
  - Study accession or unique name (alias)
  - Sample accession or unique name (alias)
- Experiment name
- Sequencing platform
- Sequencing instrument
- Library name (optional)
- Library source
- Library selection
- Library strategy
- Free text library description (optional)
- Insert size for paired reads (optional)

- Read data file(s)
  - BAM file
  - CRAM file
  - Single Fastq file
  - Paired Fastq files

**Prepare the files**

The set of files that are part of the submission are specified using a manifest file. The manifest file is specified using the `manifest <filename>` option.

A sequence read submission consists of the following files:

- 1 manifest file
- 1 BAM file, 1 CRAM file, or 1-2 Fastq files
Manifest file

The manifest file has two columns separated by a tab (or any whitespace characters):

• Field name (first column): case insensitive field name
• Field value (second column): field value

The following metadata fields are supported in the manifest file:

• STUDY: Study accession or unique name (alias)
• SAMPLE: Sample accession or unique name (alias)
• NAME: Unique experiment name
• PLATFORM: See permitted values. Not needed if INSTRUMENT is provided.
• INSTRUMENT: See permitted values
• INSERT_SIZE: Insert size for paired reads
• LIBRARY_NAME: Library name (optional)
• LIBRARY_SOURCE: See permitted values
• LIBRARY_SELECTION: See permitted values
• LIBRARY_STRATEGY: See permitted values
• DESCRIPTION: free text library description (optional)

The following file name fields are supported in the manifest file:

• BAM: Single BAM file
• CRAM: Single CRAM file
• FASTQ: Single fastq file

For example, the following manifest file represents a paired Fastq submission:

```
STUDY TODO
SAMPLE TODO
NAME TODO
INSTRUMENT Illumina Genome Analyzer II
INSERT_SIZE 200
LIBRARY_SOURCE GENOMIC
LIBRARY_SELECTION RANDOM
LIBRARY_STRATEGY WGS
FASTQ read1.fastq.gz
FASTQ read2.fastq.gz
```

Metadata validation

Permitted values for platform

• LS454: 454 technology use 1-color sequential flows
• ILLUMINA: Illumina is 4-channel flowgram with 1-to-1 mapping between basecalls and flows
• PACBIO_SMRT: PacificBiosciences platform type for the single molecule real time (SMRT) technology.
• ION_TORRENT: Ion Torrent Personal Genome Machine (PGM) from Life Technologies.
• CAPILLARY: Sequencers based on capillary electrophoresis technology manufactured by LifeTech (formerly Applied BioSciences).
• OXFORD_NANOPORE: Oxford Nanopore platform type. nanopore-based electronic single molecule analysis.
• BGISEQ
• DNBSEQ

**Permitted values for instrument**

• 454 GS
• 454 GS 20
• 454 GS FLX
• 454 GS FLX+
• 454 GS FLX Titanium
• 454 GS Junior
• HiSeq X Five
• HiSeq X Ten
• Illumina Genome Analyzer
• Illumina Genome Analyzer II
• Illumina Genome Analyzer IIx
• Illumina HiScanSQ
• Illumina HiSeq 1000
• Illumina HiSeq 1500
• Illumina HiSeq 2000
• Illumina HiSeq 2500
• Illumina HiSeq 3000
• Illumina HiSeq 4000
• Illumina iSeq 100
• Illumina MiSeq
• Illumina MiniSeq
• Illumina NovaSeq 6000
• NextSeq 500
• NextSeq 550
• PacBio RS
• PacBio RS II
• Sequel
• Ion Torrent PGM
• Ion Torrent Proton
• Ion Torrent S5
• Ion Torrent S5 XL
• AB 3730xL Genetic Analyzer
• AB 3730 Genetic Analyzer
• AB 3500xL Genetic Analyzer
• AB 3500 Genetic Analyzer
• AB 3130xL Genetic Analyzer
• AB 3130 Genetic Analyzer
• AB 310 Genetic Analyzer
• MinION
• GridION
• PromethION
• BGISEQ-500
• DNBSEQ-T7
• DNBSEQ-G400
• DNBSEQ-G50
• DNBSEQ-G400 FAST
• unspecified

Permitted values for library selection

• RANDOM: No Selection or Random selection
• PCR: target enrichment via PCR
• RANDOM PCR: Source material was selected by randomly generated primers.
• RT-PCR: target enrichment via
• HMPR: Hypo-methylated partial restriction digest
• MF: Methyl Filtrated
• repeat fractionation: Selection for less repetitive (and more gene rich) sequence through Cot filtration (CF) or other fractionation techniques based on DNA kinetics.
• size fractionation: Physical selection of size appropriate targets.
• MSLL: Methylation Spanning Linking Library
• cDNA: PolyA selection or enrichment for messenger RNA (mRNA); synonymize with PolyA
• cDNA_randomPriming:
• cDNA_oligo_dT:
• PolyA: PolyA selection or enrichment for messenger RNA (mRNA); should replace cDNA enumeration.
• Oligo-dT: enrichment of messenger RNA (mRNA) by hybridization to Oligo-dT.
• Inverse rRNA: depletion of ribosomal RNA by oligo hybridization.
• Inverse rRNA selection: depletion of ribosomal RNA by inverse oligo hybridization.

• ChIP: Chromatin immunoprecipitation

• ChIP-Seq: Chromatin immunoPrecipitation, reveals binding sites of specific proteins, typically transcription factors (TFs) using antibodies to extract DNA fragments bound to the target protein.

• MNase: Identifies well-positioned nucleosomes. uses Micrococcal Nuclease (MNase) is an endo-exonuclease that processively digests DNA until an obstruction, such as a nucleosome, is reached.

• DNase: DNase I endonuclease digestion and size selection reveals regions of chromatin where the DNA is highly sensitive to DNase I.

• Hybrid Selection: Selection by hybridization in array or solution.

• Reduced Representation: Reproducible genomic subsets, often generated by restriction fragment size selection, containing a manageable number of loci to facilitate re-sampling.

• Restriction Digest: DNA fractionation using restriction enzymes.

• 5-methylcytidine antibody: Selection of methylated DNA fragments using an antibody raised against 5-methylcytosine or 5-methylcytidine (m5C).

• MBD2 protein methyl-CpG binding domain: Enrichment by methyl-CpG binding domain.

• CAGE: Cap-analysis gene expression.

• RACE: Rapid Amplification of cDNA Ends.

• MDA: Multiple Displacement Amplification, a non-PCR based DNA amplification technique that amplifies a minute quantifies of DNA to levels suitable for genomic analysis.

• padlock probes capture method: Targeted sequence capture protocol covering an arbitrary set of nonrepetitive genomics targets. An example is capture bisulfite sequencing using padlock probes (BSPP).

• other: Other library enrichment, screening, or selection process.

• unspecified: Library enrichment, screening, or selection is not specified.

Permitted values for library source

• GENOMIC: Genomic DNA (includes PCR products from genomic DNA).

• GENOMIC SINGLE CELL:

• TRANSCRIPTOMIC: Transcription products or non genomic DNA (EST, cDNA, RT-PCR, screened libraries).

• TRANSCRIPTOMIC SINGLE CELL:

• METAGENOMIC: Mixed material from metagenome.

• METATRANSCRIPTOMIC: Transcription products from community targets

• SYNTHETIC: Synthetic DNA.

• VIRAL RNA: Viral RNA.

• OTHER: Other, unspecified, or unknown library source material.

Permitted values for library strategy

• WGS: Whole Genome Sequencing - random sequencing of the whole genome (see pubmed 10731132 for details)
• WGA: Whole Genome Amplification followed by random sequencing. (see pubmed 1631067,8962113 for details)
• WXS: Random sequencing of exonic regions selected from the genome. (see pubmed 20111037 for details)
• RNA-Seq: Random sequencing of whole transcriptome, also known as Whole Transcriptome Shotgun Sequencing, or WTSS). (see pubmed 18611170 for details)
• ssRNA-seq: Strand-specific RNA sequencing.
• miRNA-Seq: Micro RNA sequencing strategy designed to capture post-transcriptional RNA elements and include non-coding functional elements. (see pubmed 21787409 for details)
• ncRNA-Seq: Capture of other non-coding RNA types, including post-translation modification types such as snRNA (small nuclear RNA) or snoRNA (small nucleolar RNA), or expression regulation types such as siRNA (small interfering RNA) or piRNA/piwi/RNA (piwi-interacting RNA).
• FL-cDNA: Full-length sequencing of cDNA templates
• EST: Single pass sequencing of cDNA templates
• Hi-C: Chromosome Conformation Capture technique where a biotin-labeled nucleotide is incorporated at the ligation junction, enabling selective purification of chimeric DNA ligation junctions followed by deep sequencing.
• ATAC-seq: Assay for Transposase-Accessible Chromatin (ATAC) strategy is used to study genome-wide chromatin accessibility. alternative method to DNase-seq that uses an engineered Tn5 transposase to cleave DNA and to integrate primer DNA sequences into the cleaved genomic DNA.
• WCS: Random sequencing of a whole chromosome or other replicon isolated from a genome.
• RAD-Seq:
• CLONE: Genomic clone based (hierarchical) sequencing.
• POOLCLONE: Shotgun of pooled clones (usually BACs and Fosmids).
• AMPLICON: Sequencing of overlapping or distinct PCR or RT-PCR products. For example, metagenomic community profiling using SSU rRNA.
• CLONEEND: Clone end (5’, 3’, or both) sequencing.
• FINISHING: Sequencing intended to finish (close) gaps in existing coverage.
• ChIP-Seq: ChIP-seq, Chromatin ImmunoPrecipitation, reveals binding sites of specific proteins, typically transcription factors (TFs) using antibodies to extract DNA fragments bound to the target protein.
• MNase-Seq: Identifies well-positioned nucleosomes. uses Micrococcal Nuclease (MNase) is an endonuclease that processively digests DNA until an obstruction, such as a nucleosome, is reached.
• DNase-Hypersensitivity: Sequencing of hypersensitive sites, or segments of open chromatin that are more readily cleaved by DNaseI.
• Bisulfite-Seq: MethylC-seq. Sequencing following treatment of DNA with bisulfite to convert cytosine residues to uracil depending on methylation status.
• CTS: Concatenated Tag Sequencing
• MRE-Seq: Methylation-Sensitive Restriction Enzyme Sequencing.
• MeDIP-Seq: Methylated DNA Immunoprecipitation Sequencing.
• MBD-Seq: Methyl CpG Binding Domain Sequencing.
• Tn-Seq: Quantitatively determine fitness of bacterial genes based on how many times a purposely seeded transposon gets inserted into each gene of a colony after some time.
• VALIDATION: CGHub special request: Independent experiment to re-evaluate putative variants.
• FAIRE-seq: Formaldehyde Assisted Isolation of Regulatory Elements. Reveals regions of open chromatin.
• SELEX: Systematic Evolution of Ligands by Exponential enrichment
• RIP-Seq: Direct sequencing of RNA immunoprecipitates (includes CLIP-Seq, HITS-CLIP and PAR-CLIP).
• ChIA-PET: Direct sequencing of proximity-ligated chromatin immunoprecipitates.
• Synthetic-Long-Read: binning and barcoding of large DNA fragments to facilitate assembly of the fragment
• Targeted-Capture: Enrichment of a targeted subset of loci.
• OTHER: Chromatin Conformation Capture:
• OTHER: Library strategy not listed.

CRAM file validation

Reference sequence validation

Reference sequences in CRAM files are required to exists in ENA’s CRAM reference registry.
The Webin command line submission interface maintains two file based caches to avoid unnecessary calls to the registry:

1. Cache for reference sequence checksums
2. Cache for reference sequences

The cache for reference sequence checksums is stored in the $HOME/.webin-cli/cram-ref-info directory, where $HOME is the home directory for the user executing the program.

The cache for reference sequences is configured using the REF_PATH and REF_CACHE environmental variables as in samtools.

5.3.3 Submit Raw Reads Programmatically

Sequence read data is submitted using experiment and run XMLs.

An experiment object represents the library solution that is created from a sample and used in a sequencing experiment. The experiment object contains details about the sequencing platform and library protocols.

A run object represents a lane (or equivalent) on an sequencing machine and is used to attach sequence read data to experiments.

The experiment XML format is defined by SRA.experiment.xsd XML Schema, and the run XML format is defined by SRA.run.xsd XML Schema.

Object relationships

Both run and experiment are associated with other objects.
An experiment is part of a study. Studies are used to group together experiments to allow them to be cited together in a publication.

An experiment is associated with a sample. It is common to have multiple libraries and sequencing experiments for a single sample. Experiments point to samples to allow sharing of sample information between multiple experiments.

To summarise object relationships:

1. One or more runs are part of an experiment.
2. One or more experiments are part of a study.
3. One or more experiments are associated with a sample.

It is common to pre-register samples ahead of submitting sequence reads. Note that samples and studies will be associated with each only after experiments have been submitted.

**Run XML: part of experiment**

A run points to the experiment it is part of using the `<EXPERIMENT_REF>` element. This can be done either by using an accession:

```xml
<EXPERIMENT_REF accession="ERX123456"/>
```

or a name within the submitter’s account:

```xml
<EXPERIMENT_REF refname="exp_mantis_religiosa"/>
```

Above, the `refname` refers to the submitter provided name (alias) of the experiment.

If the run is submitted at the same time as the experiment then the `accession` attribute can’t be used to refer to the experiment as the experiment accession has not been assigned yet. Had the experiment been submitted previously then the `accession` attribute could have been used.
When referring to experiments in other submission accounts, for example in a collaborators account, the *accession* attribute must be used. Names can be ambiguous between submission accounts as many other submitted objects could share the same name.

These principles with *refname* and *accession* attributes applies to all references between objects including experiment references to studies and samples.

**Experiment XML: part of study**

An experiment points to the study it is part of using the `<STUDY_REF>` element. In the example below this reference is made using an accession number:

```xml
<STUDY_REF accession="ERP123456"/>
```

Either an *accession* or a *refname* (alias) can be used in the reference. If you are using the *accession* attribute you can use both ERP and PRJ accessions when referring to studies.

**Experiment XML: associated with sample**

An experiment points to the sample it is associated with using the `<SAMPLE_DESCRIPTOR>` element. In the example below this reference is made using an accession number:

```xml
<SAMPLE_DESCRIPTOR accession="SRS462875"/>
```

Either an *accession* or a *refname* (alias) can be used in the reference. If you are using the *accession* attribute you can use both ERS and SAM accessions when referring to samples.

**Metadata standards**

The ENA is involved with minimum information standards for various project collaborations and sequencing fields. Even if you are not a member of one of these you can still use them as a guideline for increasing the quality of annotation of your experiment (and run) objects towards more interpretable and reproducible publications.

**Supported data formats**

Please see Read Data Formats.

**Upload data files**

Please see Data Upload.

You must have uploaded data files into your Webin upload area before you can submit them using a run XML. Once the run has been submitted the data files will be moved from the Webin upload area into the archive.

You can upload your data files to the root directory of your upload area or you can create subdirectories and upload your files there.

If the files are uploaded to the root directory then simply enter the file name in the Run XML when referring to it:

```xml
<FILE filename="mantis_religiosa_R1.fastq.gz" ... />
```

If the files are uploaded into a subdirectory (e.g. *mantis_religiosa*) then prefix the file name with the name of the subdirectory:
Create the Run and Experiment XML

Run and Experiment XML: paired fastq

Below is an example of an Illumina HiSeq 2000 paired end reads being submitted in Fastq format.

The experiment points to a pre-registered sample and study using their accessions.

The run points to the experiment using the experiment’s alias.

Experiment XML:

```xml
<EXPERIMENT_SET>
  <EXPERIMENT alias="exp_mantis_religiosa">
    <TITLE>The 1KITE project: evolution of insects</TITLE>
    <STUDY_REF accession="SRP017801"/>
    <DESIGN>
      <LIBRARY_DESCRIPTOR>
        <LIBRARY_NAME/>
        <LIBRARY_STRATEGY>RNA-Seq</LIBRARY_STRATEGY>
        <LIBRARY_SOURCE>TRANSCRIPTOMIC</LIBRARY_SOURCE>
        <LIBRARY_SELECTION>cDNA</LIBRARY_SELECTION>
        <LIBRARY_LAYOUT>
          <PAIRED NOMINAL_LENGTH="250" NOMINAL_SDEV="30"/>
        </LIBRARY_LAYOUT>
        <LIBRARY_CONSTRUCTION_PROTOCOL>
          Messenger RNA (mRNA) was isolated using the Dynabeads mRNA Purification Kit (Invitrogen, Carlsbad Ca. USA) and then sheared using divalent cations at 72°C. These cleaved RNA fragments were transcribed into first-strand cDNA using II Reverse Transcriptase (Invitrogen, Carlsbad Ca. USA) and N6 primer (IDT). The second-strand cDNA was subsequently synthesized using RNase H (Invitrogen, Carlsbad Ca. USA) and DNA polymerase I (Invitrogen, Shanghai China). The double-stranded cDNA then underwent end-repair, a single 'A' base addition, adapter ligatiation, and size selection on anagarose gel (250 * 20 bp). At last, the product was indexed and PCR amplified to finalize the library preparation for the paired-end cDNA. </LIBRARY_CONSTRUCTION_PROTOCOL>
      </LIBRARY_DESCRIPTOR>
    </DESIGN>
    <PLATFORM>
      <ILLUMINA>
        <INSTRUMENT_MODEL>Illumina HiSeq 2000</INSTRUMENT_MODEL>
      </ILLUMINA>
    </PLATFORM>
    <EXPERIMENT_ATTRIBUTES>
      <EXPERIMENT_ATTRIBUTE>
        <TAG>library preparation date</TAG>
        <VALUE>2010-08</VALUE>
      </EXPERIMENT_ATTRIBUTE>
    </EXPERIMENT_ATTRIBUTES>
  </EXPERIMENT>
</EXPERIMENT_SET>
```

Run XML: 5.3. Submission Options
You can submit several experiments and runs at the same time by using multiple `<EXPERIMENT>` and `<RUN>` blocks.

Experiment XML:

```xml
<EXPERIMENT_SET>
  <EXPERIMENT alias="exp_01">
    ...
  </EXPERIMENT>
  ...
  <EXPERIMENT alias="exp_05">
    ...
  </EXPERIMENT>
</EXPERIMENT_SET>
```

Run XML:

```xml
<RUN_SET>
  <RUN alias="run_01">
    <EXPERIMENT_REF refname="exp_01"/>
    <DATA_BLOCK>
      <FILES>
        ...
      </FILES>
    </DATA_BLOCK>
  </RUN>
  ...
  <RUN alias="run_05">
    <EXPERIMENT_REF refname="exp_05"/>
    <DATA_BLOCK>
      <FILES>
        ...
      </FILES>
    </DATA_BLOCK>
  </RUN>
</RUN_SET>
```

Change the XMLs by entering your own information and save it in two files, for example `experiment.xml` and `run.xml`.

Change the value of `alias` to be a unique name. You will need the unique name for example to refer to your experiment when adding run objects to it. An alias can be a short acronym but it should be meaningful and memorable in some way.
The <FILES> block in the run XML references the data files that are being submitted as part of the run.

To check the integrity of the file transfer an md5 checksum must be provided for each file. You can provide this by using the checksum_method="MD5" and checksum attributes in the <FILE> element, or you can provide the MD5 checksum in file <file>.md5 in the same folder as the corresponding data file <file>.

**Run XML: BAM**

Below is an example of a RUN XML when reads are submitted in BAM format. Please note that filetype has been set to bam.

```xml
<RUN_SET>
  <RUN alias="run_mantis_religiosa" center_name="">
    <EXPERIMENT_REF refname="exp_run_mantis_religiosa"/>
    <DATA_BLOCK>
      <FILES>
        <FILE filename="mantis_religiosa_R1.bam" filetype="bam"
              checksum_method="MD5" checksum="9b8932f85caa54e687eba62fca3edce2"/>
      </FILES>
    </DATA_BLOCK>
  </RUN>
</RUN_SET>
```

**Run XML: CRAM**

Below is an example of a RUN XML when reads are submitted in CRAM format. Please note that filetype has been set to cram.

```xml
<RUN_SET>
  <RUN alias="run_mantis_religiosa" center_name="">
    <EXPERIMENT_REF refname="exp_run_mantis_religiosa"/>
    <DATA_BLOCK>
      <FILES>
        <FILE filename="mantis_religiosa_R1.cram" filetype="cram"
              checksum_method="MD5" checksum="9b8932f85caa54e687eba62fca3edce2"/>
      </FILES>
    </DATA_BLOCK>
  </RUN>
</RUN_SET>
```

**Run XML: Oxford Nanopore**

If you wish to submit your nanopore sequencing reads in their native FAST5 format, you must prepare an individual gzipped tar archive for each run. The run XML should look as follows:

```xml
<RUN_SET>
  <RUN alias="run_mantis_religiosa" center_name="">
    <EXPERIMENT_REF refname="exp_run_mantis_religiosa"/>
    <DATA_BLOCK>
      <FILES>
        <FILE filename="exp_run_mantis_religiosa.tar.gz" filetype="OxfordNanopore_native"/>
      </FILES>
    </DATA_BLOCK>
  </RUN>
</RUN_SET>
```
Experiment XML: library information

The experiment contains a `<LIBRARY_DESCRIPTOR>` block in order to capture basic library information within the `<LIBRARY_STRATEGY>`, `<LIBRARY_SOURCE>` and `<LIBRARY_SELECTION>` elements. These are controlled value fields and the permitted values are listed in the SRA.experiment.xsd XML Schema.

...<LIBRARY_DESCRIPTOR>
  <LIBRARY_NAME/>
  <LIBRARY_STRATEGY>RNA-Seq</LIBRARY_STRATEGY>
  <LIBRARY_SOURCE>TRANSCRIPTOMIC</LIBRARY_SOURCE>
  <LIBRARY_SELECTION>cDNA</LIBRARY_SELECTION>
  <LIBRARY_LAYOUT>
    <PAIRED NOMINAL_LENGTH="250" NOMINAL_SDEV="30"/>
  </LIBRARY_LAYOUT>
  <LIBRARY_CONSTRUCTION_PROTOCOL>
    Messenger RNA (mRNA) was isolated using the Dynabeads mRNA Purification Kit (Invitrogen, Carlsbad Ca. USA) and then sheared using divalent cations at 72°C. These cleaved RNA fragments were transcribed into first-strand cDNA using SuperScript?II Reverse Transcriptase (Invitrogen, Carlsbad, Ca. USA) and N6 primer (IDT). The second-strand cDNA was subsequently synthesized using RNase H (Invitrogen, Carlsbad Ca. USA) and DNA polymerase I (Invitrogen, Shanghai China). The double-stranded cDNA then underwent end-repair, a single `A`-base addition, adapter ligation, and size selection on agarose gel (250 * 20 bp). At last, the product was indexed and PCR amplified to finalize the library preparation for the paired-end cDNA.
  </LIBRARY_CONSTRUCTION_PROTOCOL>
</LIBRARY_DESCRIPTOR>
...

Experiment XML: paired ended reads

In experiment XML, the `<LIBRARY_LAYOUT>` element within the `<LIBRARY_DESCRIPTOR>` differentiates between single and paired ended reads. In the Experiment XML fragment below, we have described a paired ended experiment using the `<PAIRED>` element:

...<LIBRARY_LAYOUT>
  <PAIRED NOMINAL_LENGTH="250" NOMINAL_SDEV="30"/>
</LIBRARY_LAYOUT>
...

The attribute `PAIRED NOMINAL_LENGTH` is the average insert size. It is not the length of the reads. It is the average size of the fragments that are being sequenced.

The attribute `NOMINAL_SDEV` is the standard deviation of the fragment lengths. This attribute is not mandatory so you can omit it if you do not have this detail.
Experiment XML: single ended reads

In the Experiment XML fragment below, we have described a single ended experiment using the \texttt{<SINGLE>} element:

```
...  
<LIBRARY_LAYOUT>
  <SINGLE/>
</LIBRARY_LAYOUT>
...  
```

Experiment and Run XML: attributes

Additional annotation can be provided for experiments and runs using attributes in the XMLs.

In experiment XML these attributes are captured using the \texttt{EXPERIMENT_ATTRIBUTE} block:

```
<EXPERIMENT_ATTRIBUTE>
  <TAG>library preparation date</TAG>
  <VALUE>2010-08</VALUE>
</EXPERIMENT_ATTRIBUTE>
```

In run XML these attributes are captured using the \texttt{RUN_ATTRIBUTE} block which is identical to the \texttt{EXPERIMENT_ATTRIBUTE} except for the name.

Annotations using attribute blocks should fit the type of object. For an experiment object you would expect annotations related to the library solution and for a run object you would expect annotations related to the NGS machine run.

Please refer to the metadata standards section in this module for information about recommended experiment and run attributes.

Create the Submission XML

To submit experiments or runs, you need an accompanying submission XML in a separate file. Let’s call this file \texttt{submission.xml}.

```
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <ADD/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```

The submission XML declares one or more Webin submission service actions. In this case the action is \texttt{<ADD/>} which is used to submit new objects.

The XMLs can be submitted programmatically, using CURL on command line or using the Webin submissions portal.

Submit the XMLs using CURL

CURL is a Linux/Unix command line program which you can use to send the \texttt{experiment.xml}, \texttt{run.xml} and \texttt{submission.xml} to the Webin submission service.

```
curl -u username:password -F "SUBMISSION=@submission.xml" -F "EXPERIMENT=@experiment.xml" -F "RUN=@run.xml" "https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit/"
```
Please provide your Webin submission account credentials using the username and password.

After running the command above a receipt XML is returned. It will look like the one below:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<RECEIPT receiptDate="2017-08-11T15:07:36.746+01:00" submissionFile="sub.xml" success="true">
  <EXPERIMENT accession="ERX2151578" alias="exp_mantis_religiosa" status="PRIVATE"/>
  <RUN accession="ERR2094164" alias="run_mantis_religiosa" status="PRIVATE"/>
  <SUBMISSION accession="ERA986371" alias="mantis_religiosa_submission"/>
  <MESSAGES>
    <INFO>This submission is a TEST submission and will be discarded within 24 hours</INFO>
  </MESSAGES>
  <ACTIONS>ADD</ACTIONS>
  <ACTIONS>ADD</ACTIONS>
</RECEIPT>
```

**Submit the XMLs using Webin submissions portal**

XMLs can also be submitted interactively using the Webin submissions portal. Please refer to the Webin submissions portal document for an example how to submit a study using XML. Other types of XMLs can be submitted using the same approach.

**The Receipt XML**

To know if the submission was successful look in the first line of the `<RECEIPT>` block.

The attribute success will have value true or false. If the value is false then the submission did not succeed. In this case check the rest of the receipt for error messages and after making corrections, try the submission again.

If the success attribute is true then the submission was successful. The receipt will contain the accession numbers of the objects that you have submitted.

**Test and production services**

Note the message in the receipt:

```
<INFO>This submission is a TEST submission and will be discarded within 24 hours</INFO>
```

It is advisable to first test your submissions using the Webin test service where changes are not permanent and are erased every 24 hours.

Once you are happy with the result of the submission you can use the CURL command again but this time using the production service. Simply change the part in the URL from `wwwdev.ebi.ac.uk` to `www.ebi.ac.uk`:

```
curl -u username:password -F "SUBMISSION=@submission.xml" -F "EXPERIMENT=@experiment.xml" -F "RUN=@run.xml" "https://www.ebi.ac.uk/ena/submit/drop-box/submit/
```

Similarly, if you are using the Webin submissions portal change the URL from `wwwdev.ebi.ac.uk` to `www.ebi.ac.uk`.

102 Chapter 5. How to Submit Raw Reads
6.1 Introduction

To submit genome or transcriptome assemblies to ENA you must also provide some metadata to describe your research project. This helps make your data re-useable and searchable.

Within ENA, all assemblies are submitted as ‘analysis’ submission objects but are processed differently depending on what type of assembly is submitted.

If you are not yet familiar with the metadata model, please see [here](#) for some more information.
As an assembly references ENA sample and study objects, you must submit these before you submit your data. It is also strongly recommended to submit as well as reference any reads associated with the assembly being submitted.

See below for information on how to: register a study within ENA to describe your overall research project, register samples with information on the biological data that was sequenced then assembled and submit any reads associated with each sample being submitted.

- Register a study
- Register a sample
- Submit raw reads

### 6.2 Assembly Levels

Before submitting your assembly, consider the highest level of assembly which has been attained. This will have implications for how you prepare your submission, as well as the accessions you receive at the end.

ENA recognises three assembly levels:

- **Contig**: the highest level of assembly is contigs
- **Scaffold**: the highest level of assembly consists of gapped contigs (scaffolds)
- **Chromosome**: the highest level of assembly includes assembled chromosomes

Note that ‘chromosome’ should here be understood as a general term for a range of complete replicons, including chromosomes of eukaryotes, prokaryotes, and viruses, as well as organellar chromosomes and plasmids. All of these may be submitted within the same chromosome-level assembly.

Please note that contig and scaffold level assemblies can both be updated to higher level assemblies after submission. You cannot update to a lower level assembly, however, and you cannot add functional annotation if none was present in the first submission.

### 6.3 Files For Genome Assembly Submissions

File requirements for a genome assembly submission depends on the assembly level and are specified using a manifest file. The set of files required for genome assembly submissions are listed in the following table:
### 6.4 Accessions

As all assemblies in ENA are submitted as ‘analyses’, for each assembly submission, Webin will report a unique accession number that starts with ERZ. For most assemblies, this accession number is for internal processing only and will not be visible in the browser. As a result, for most assemblies you will receive additional post-processing accession numbers starting with GCA_.

In alignment with INSDC partners, SARS-CoV-2 assemblies will not be assigned a GCA_ accession. For these assemblies, sequence accessions will continue to be assigned and the ERZ records will also be available in the browser to provide a point of access for the submitted file(s).

Please see individual submission guidelines for information on what accessions you will receive for each assembly type.

Always make a note of any accessions you receive as these are the unique identifiers for each of your submissions to ENA.

The ERZ accession can be used to access information on the progress of the internal processing of each assembly through the Webin Submissions Portal or Webin Reports Service.

### 6.5 Submission Options

Genome and transcriptome assemblies can only be submitted using the Webin-CLI submission interface. For an overview of how to use this, please see the documentation on Webin-CLI Submission.

#### 6.5.1 Submitting Genome Assemblies of Individuals or Cultured Isolates

- **Introduction**
- **Stage 1: Pre-Register Study And Sample**
- **Stage 2: Prepare The Files**
• Stage 3: Validate And Submit The Files
• Assigned Accession Numbers
• Validation Rules
• Assembly Updates
• Automatic Fixes

Introduction

Genome assemblies can be submitted to the European Nucleotide Archive (ENA) using the Webin command line submission interface with `-context genome` option.

Please contact our helpdesk if you intend to submit an assembly assembled from third party data.

Genome assembly submissions include plasmids, organelles, complete virus genomes, viral segments/replicons, bacteriophages, prokaryotic and eukaryotic genomes.

A genome assembly submission includes:

• General assembly information
  – Study accession or unique name (alias)
  – Sample accession or unique name (alias)
  – Assembly name
  – Assembly type
  – Assembly program
  – Sequencing platform
  – Minimum gap length
  – Molecule type (genomic DNA, genomic RNA or viral cRNA)
  – Coverage
  – Free text description of the genome assembly (optional)

• Contig sequences (if any)
• Scaffold sequences (if any)
• Chromosome sequences (if any)
• Unlocalised sequences (if any)
• Functional annotation (optional)

For assembly submission purposes, the term ‘chromosome’ should be understood to include organelles (e.g. mitochondria and chloroplasts), plasmids and viral segments.

The below image provides an outline of the workflow for submitting assemblies:
Stage 1: Pre-Register Study And Sample

Each submission must be associated with a pre-registered study and a sample.

Genome assemblies except primary metagenomes are uniquely associated with a study and a sample. When assemblies are updated they must be re-submitted with the same study and sample as in the original submission.

- Register a Study
- Register a Sample

It is also strongly recommended to submit the reads from which the assembly was assembled. In order to reference the reads which were used to generate the assembly, please see the `RUN_REF` tag included in the manifest file below.

- Submit raw reads

Register locus tag prefixes

This is only required if you are submitting an annotated assembly. Otherwise, please proceed to Stage 2.

Locus tag prefixes should be registered with your study. See here for information on locus tags. Briefly, locus tag prefixes must:

- Start with a letter
- Be at least 3 characters long
- Be upper case
- Contain only alpha-numeric characters and no symbols such as _-

You will need to wait 24 hours before proceeding further in the submission process to ensure these prefixes have been registered.

Stage 2: Prepare The Files

The set of files that are part of the submission are specified using a manifest file. The manifest file is identified in the Webin-CLI command using the `manifest <filename>` option.
Please note that the types of file which may be involved in an assembly submission are more comprehensively documented in our page Accepted Genome Assembly Data Formats.

The set of files required for submission of a genome assembly depends on the assembly level:

- **Contig Assembly**
  - 1 manifest file
  - 1 FASTA file OR 1 flat file

This assembly level only requires information on the sequences and annotation (if any). You will receive an error if less than 2 or more than 1,000,000 sequences are submitted. If you have less than 2 sequences, then you will need to submit at a higher assembly level or as template sequences. If you have more than 1,000,000 contigs in your submission, please contact the helpdesk.

See an example contig-level assembly at: [https://www.ebi.ac.uk/ena/browser/view/GCA_000003085](https://www.ebi.ac.uk/ena/browser/view/GCA_000003085)

- **Scaffold Assembly**
  - 1 manifest file
  - 1 FASTA file OR 1 flat file
  - 0-1 AGP files

This assembly level requires information on the sequences and annotation (if any). It also allows the submitter to provide an AGP file to give instructions for the assembly of the scaffolds from the contigs.

See an example scaffold-level assembly at: [https://www.ebi.ac.uk/ena/browser/view/GCA_902705575](https://www.ebi.ac.uk/ena/browser/view/GCA_902705575)

- **Chromosome Assembly**
  - 1 manifest file
  - 1 FASTA file OR 1 flat file
  - 1 chromosome list file
  - 0-1 unlocalised list files
  - 0-1 AGP files

This assembly level allows the submission of fully assembled chromosomes including organelles, plasmids, and viral segments. This requires information on the sequences and annotation (if any) and submission of a chromosome list file to indicate which sequences represent which 'chromosomes'.

If these chromosomes contain unlocalised sequences (where the chromosome of the sequence is known but not the exact location) you can submit an additional unlocalised list file. However, please note, if you wish to submit unplaced
contigs or unplaced scaffolds (with valid biological evidence), you will have to submit these at the appropriate lower level and use an AGP file to indicate which scaffolds/contigs are assembled to form each chromosome. Any sequences that are not used to assemble chromosomes are considered unplaced. Note that all sequences should still be submitted in a single FASTA or flat file. Artificial constructs without biological evidence, such as artificial chromosomes consisting of unplaced contigs or scaffolds, are not permitted to be submitted.

See an example chromosome level assembly at: https://www.ebi.ac.uk/ena/browser/view/GCA_000237925

For this assembly level in particular, it is important to understand how sequence names are formatted so they can be consistent between files, otherwise the system will just register your submission at contig level.

### Sequence Names

Sequences must have a unique name within the submission that is provided in the fasta, AGP or flat files. It is essential that the sequence names are unique and used consistently between files.

For example, the chromosome list file must refer to the chromosome sequences using the unique sequence names. Similarly, an AGP file must refer to scaffolds or contigs using the unique sequence names.

### Manifest Files

The manifest file describes your assembly, including metadata and file names. It is a plain text file with two columns separated by a tab (or any whitespace characters):

- Field name (first column): case insensitive field name
- Field value (second column): field value

The following metadata fields are supported in the manifest file for genome context:

- **STUDY**: Study accession - mandatory
- **SAMPLE**: Sample accession - mandatory
- **ASSEMBLYNAME**: Unique assembly name - mandatory
- **ASSEMBLY_TYPE**: ‘clone or isolate’ - mandatory
- **COVERAGE**: The estimated depth of sequencing coverage - mandatory
- **PROGRAM**: The assembly program - mandatory
- **PLATFORM**: The sequencing platform, or comma-separated list of platforms - mandatory
- **MINGAPLENGTH**: Minimum length of consecutive Ns to be considered a gap - optional
- **MOLECULETYPE**: ‘genomic DNA’, ‘genomic RNA’ or ‘viral cRNA’ - optional
- **DESCRIPTION**: Free text description of the genome assembly - optional
- **RUN_REF**: Comma separated list of run accession(s) - optional

Please see further below for validation rules affecting some of these fields.

Various file name fields are supported in the manifest file. Note that all of these are optional, though of course at least one must be provided, and some may only be relevant in the presence of other file types. The available fields are as follows:

- **FASTA**: sequences in fasta format
- **FLATFILE**: sequences in EMBL-Bank flat file format
- **AGP**: sequences in AGP format

---

6.5. Submission Options 109
• CHROMOSOME_LIST: list of chromosomes

• UNLOCALISED_LIST: list of unlocalised sequences

For example, the following manifest file represents a genome assembly consisting of contigs provided in one fasta file:

```
STUDY     TODO
SAMPLE    TODO
ASSEMBLYNAME  TODO
ASSEMBLY_TYPE  clone or isolate
COVERAGE     TODO
PROGRAM      TODO
PLATFORM     TODO
MINAPLENGTH  TODO
MOLECULETYPE genomic DNA
FASTA       genome.fasta.gz
```

Stage 3: Validate And Submit The Files

Files are validated, uploaded and submitted using the Webin command line submission interface (Webin-CLI). Please refer to the Webin command line submission interface documentation for more information about the submission process.

Assigned Accession Numbers

Once the genome assembly has been submitted an analysis (ERZxxxxxx) accession number is immediately assigned and returned to the submitter by the Webin command line submission interface (Webin-CLI).

ERZ accessions should not be used to reference the assembly in publications. The purpose of the ERZ accession number is for the submitter to be able to refer to their submission within the Webin submission service. For example, the submitter can retrieve the assigned genome assembly and sequence accessions from the Webin submissions portal or from the Webin reports service using the ERZ accession number. This accession should be used to refer to the assembly in any conversations with helpdesk staff.

For genome assemblies, long term stable accession numbers that can be used in publications are:

• Study accession (PRJEBxxxxxx) assigned at time of study registration
• Sample accession (SAMEAxxxxxx) assigned at time of study registration
• Genome assembly accession (GCA_xxxxxx) assigned once the genome assembly has been fully processed by ENA and is released on agreement with GenCol. The GCA is stable between versions
• Sequence accession(s) assigned once the genome assembly submission has been fully processed by ENA

Submitters can retrieve the genome and sequence accession numbers from the Webin submissions portal or from the Webin reports service. These accession numbers are also sent to the submitters by e-mail.

Validation Rules

Assembly submissions are subject to a great deal of validation before submission is allowed. Some key points are described here.

Sample And Study Validation

• Sample and study (BioProject) pair must be unique for an assembly (except primary metagenomes)
• Sample taxonomic classification must be species rank or below (or equivalent) within NCBI taxonomy.

Assembly Name Validation

Assembly names must:
  • match the pattern: ^[A-Za-z0-9][A-Za-z0-9 _#-.]*$
  • not be longer than 50 characters
  • not include the taxonomic name of the organism assembled

Chromosome Name Validation

Chromosome names must:
  • match the pattern: ^[A-Za-z0-9][A-Za-z0-9_#-.]*$
  • be shorter than 33 characters
  • be unique within an assembly
  • not contain any of the following as part of their name (case insensitive):
    • ‘chr’
    • ‘chrm’
    • ‘chrom’
    • ‘chromosome’
    • ‘linkage group’
    • ‘linkage-group’
    • ‘linkage_group’
    • ‘plasmid’

Sequence Validation

Sequences must:
  • have unique names within an assembly
  • be at least 20bp long
  • not have terminal Ns

Sequence Count Validation

Assembly submissions will typically not be allowed if the amount of sequences doesn’t fall within a required range. For example, contig-level assemblies must include more than 1 but less than 1,000,000 sequences. In specific cases, ENA may allow the submission of genome assemblies that are giving the following errors:

Since Webin 1.7.0:

6.5. Submission Options
• Invalid number of sequences : XXX, Minimum number of sequences for CONTIG is: YYY
• Invalid number of sequences : XXX, Minimum number of sequences for SCAFFOLD is: YYY
• Invalid number of sequences : XXX, Minimum number of sequences for CHROMOSOME is: YYY
• Invalid number of sequences : XXX, Maximum number of sequences for CONTIG is: YYY
• Invalid number of sequences : XXX, Maximum number of sequences for SCAFFOLD is: YYY
• Invalid number of sequences : XXX, Maximum number of sequences for CHROMOSOME is: YYY

This will be done at the discretion of the curation team when provided with valid reasoning, and can be requested through our helpdesk.

Note that their is no limit on the quantity of sequence data in bases, just the amount of sequences in total.

Assembly Updates

Assembly updates must: - use the same sample and study pair as was used in the initial assembly submission - not remove any chromosomes - use a unique ‘ASSEMBLY_NAME’ value in the manifest file

Automatic Fixes

Some fixes may be applied automatically, which users should be aware of. A few of these are documented below.

CDS Feature

• Feature location is made 5’ partial if the /codon_start is 2 or 3.
• Feature location is made 5’ partial if the /translation does not start with a start codon.
• Feature location is made 3’ partial if the /translation does not end with a stop codon.
• Feature location is made 3’ and 5’ partial if the location span is not a multiple of three.
• Feature location has 3’ partiality removed if the /translation ends with a stop codon.
• Feature is made /pseudo and the /translation is removed if the /translation contains internal stop codons.

6.5.2 Submitting Metagenome Assemblies

Metagenome assemblies can be submitted to the European Nucleotide Archive (ENA) as one of three options.

• Primary Metagenome : a metagenome assembly prior to binning from a sampled biome or collection of sampled biomes without attempt to separate taxa.
• Binned Metagenome : encompasses anything from a set of contigs to a complete genome assembly from a metagenomic source that has been identified as a single-taxon set.
• Metagenome-Assembled Genome (MAG) : a single-taxon assembly based on one or more binned metagenomes that has been asserted to be a close representation to an actual individual genome (that could match an already existing isolate or represent a novel isolate).

This three tiered system is designed to represent the different stages of a metagenome assembly workflow:
To best represent your data, please submit as many of the levels as you can that are applicable to your study.

**Submitting A Primary Metagenome Assembly**

- **Introduction**
- **Stage 1: Pre-register study and environmental sample**
- **Stage 2: Prepare the files**
- **Stage 3: Validate and submit the files**
- **Assigned accession numbers**
- **Validation rules**

**Introduction**

Metagenome assemblies can be submitted to the European Nucleotide Archive (ENA) using the Webin command line submission interface with `-context genome`.

Please contact our helpdesk if you intend to submit an assembly assembled from third party data.

A primary metagenome assembly is a metagenome assembly prior to binning from a sampled biome or collection of sampled biomes without attempt to separate taxa.

A primary metagenome assembly consists of:

- General assembly information
  - Study accession or unique name (alias)
  - Environmental Sample accession or unique name (alias)
  - Assembly program
– Sequencing platform
– Minimum gap length
– Molecule type (genomic DNA, genomic RNA or viral cRNA)
– Coverage
– Free text description of the assembly (optional)

• Contig sequences in fasta format

The following picture illustrates the stages of the metagenome assembly submission process:

Stage 1: Pre-register study and environmental sample

Each submission must be associated with a pre-registered study and environmental sample. This sample should be the same sample used to submit your raw reads.

When registering an environmental sample, please make sure the appropriate environmental checklist is chosen for this and an environmental taxon is used (e.g. aquatic metagenome (tax id: 1169740)).

If you do not intend to submit raw reads or a primary metagenome as part of your metagenome assembly submission, please see here for details on how to do this. If no data is associated with the environmental sample, it needs to be released manually in order to be available to the public.

The methods for submitting metagenomic studies and samples follow the same process as any other study/sample submission. Follow the links for more information.

• Register a Study
• Register a Sample

It is strongly recommended to submit as well as reference any reads associated with the assembly being submitted. In order to reference the reads which were used to generate the assembly, please see the RUN_REF tag included in the manifest file below.

• Submit raw reads
Stage 2: Prepare the files

The set of files that are part of the submission are specified using a manifest file. The manifest file is specified using the `-manifest <filename>` option.

A primary metagenome assembly submission consists of the following files:

- 1 manifest file
- 1 fasta file

Manifest file

The manifest file has two columns separated by a tab (or any whitespace characters):

- Field name (first column): case insensitive field name
- Field value (second column): field value

The following metadata fields are supported in the manifest file:

- **STUDY**: Study accession or unique name (alias)
- **SAMPLE**: Environmental sample accession or unique name (alias)
- **ASSEMBLYNAME**: Unique assembly name
- **ASSEMBLY_TYPE**: ‘primary metagenome’
- **COVERAGE**: The estimated depth of sequencing coverage
- **PROGRAM**: The assembly program
- **PLATFORM**: The sequencing platform, or comma-separated list of platforms
- **MINGAPLENGTH**: Minimum length of consecutive Ns to be considered a gap (optional)
- **MOLECULETYPE**: ‘genomic DNA’, ‘genomic RNA’ or ‘viral cRNA’ (optional)
- **DESCRIPTION**: Free text description of the genome assembly (optional)
- **RUN_REF**: Comma separated list of run accession(s) (optional)

Please see further below for validation rules affecting some of these fields.

The following file name fields are supported in the manifest file for a primary metagenome submission:

- **FASTA**: sequences in fasta format

For example, the following manifest file represents a primary metagenome assembly consisting of contigs provided in one fasta file:

```
STUDY   TODO
SAMPLE  TODO
ASSEMBLYNAME  TODO
ASSEMBLY_TYPE  primary metagenome
COVERAGE  TODO
PROGRAM  TODO
PLATFORM  TODO
MINGAPLENGTH  TODO
MOLECULETYPE  genomic DNA
FASTA    primary_metagenome.fasta.gz
```
Stage 3: Validate and submit the files

Files are validated, uploaded and submitted using the Webin command line submission interface. Please refer to the Webin command line submission interface documentation for more information about the submission process.

Assigned accession numbers

Once the genome assembly has been submitted an analysis (ERZxxxxxx) accession number is immediately assigned and returned to the submitter by the Webin command line submission interface. The purpose of the ERZ accession number is for the submitter to be able to refer to their submission within the Webin submission service and access their data in the browser.

For primary metagenome assemblies, long term stable accession numbers that can be used in publications are:

- Study accession (PRJEBxxxxx) assigned at time of study registration.
- Sample accession (SAMEAxxxxxx) assigned at time of sample registration.

See an example of a publicly available primary metagenome at: https://www.ebi.ac.uk/ena/browser/view/ERZ1091679

Validation rules

Assembly name validation

Assembly names must:
- match the pattern: ^[A-Za-z0-9][A-Za-z0-9 _#-._]*$  
- not be longer than 50 characters  
- not include the taxonomic name of the organism assembled

Sequence validation

Sequences must:
- have unique names within an assembly  
- be at least 20bp long  
- not have terminal Ns  
- consist of bases: 'a','c','g','t','u','b','d','h','k','m','n','r','s','v','w','y'

Submitting Binned Metagenome Assemblies

- Introduction  
- Stage 1: Pre-register study and metagenomic samples  
- Stage 2: Prepare the files  
- Stage 3: Validate and submit the files  
- Assigned accession numbers
• **Validation rules**

### Introduction

Metagenome assemblies can be submitted to the European Nucleotide Archive (ENA) using the Webin command line submission interface with `-context genome`. Please contact our helpdesk if you intend to submit an assembly assembled from third party data.

A **binned metagenome** assembly submission encompasses anything from a set of contigs to a complete genome assembly from a metagenomic source that has been identified as a single-taxon set.

There is no limit to the number of bins that can be submitted as part of a metagenomic study as it is recognised that the number of bins produced can be upwards of 100 000. Please submit all derived bins at this assembly level and not as Metagenome-Assembled Genomes unless they meet the required criteria.

A binned metagenome assembly consists of:

- General assembly information
  - Study accession or unique name (alias)
  - **Binned** Sample accession or unique name (alias)
  - Assembly program
  - Sequencing platform
  - Minimum gap length
  - Molecule type (genomic DNA, genomic RNA or viral cRNA)
  - Coverage
  - Free text description of the assembly (optional)
- Contig sequences (if any)
- Scaffold sequences (if any and submitting MAGs)
- Chromosome sequences (if any and submitting MAGs)

The following picture illustrates the stages of the metagenome assembly submission process:
Stage 1: Pre-register study and metagenomic samples

Each submission must be associated with a pre-registered study and a **binned** sample.

If you have not done so already, please register a study.

- Register a Study

It is recommended to submit any primary metagenomic assemblies and raw reads before submitting your binned metagenomes to help record your methods and make your data reproducible.

- Submitting Raw Reads
- Submitting A Primary Metagenome Assembly

Registering binned samples

Each **binned metagenome** assembly submission must be associated with a **binned** sample. This is because a bin is not an assembly of the whole set of raw data but an assembly derived from a smaller subset of those data. These virtual samples represents the subset of that data and hold all metadata related to the taxonomy of that subset as well as methods used to derive it.
These binned samples should be as specific in taxonomy as it can be and use the specific ENA binned metagenome checklist.

Please make sure these binned samples correctly reference the environmental sample that the bin was derived from. The environmental sample should be the same sample used to submit your raw reads and primary metagenomes.

This can be done from within the checklist using the mandatory “sample derived from” attribute. If the assembly was derived from multiple samples or runs you can list these with a comma separated list or range.

You should also reference the environmental sample in the description as one of the following:

“This sample represents a metagenomic bin from the metagenomic sample ERSXXXXX”

OR

“This sample represents a metagenomic bin from the metagenomic run ERRXXXXX”

What if I don’t intend to submit raw reads or a primary metagenome?

If you do not intend to submit raw reads or a primary metagenome as part of your metagenome assembly submission, please still register an environmental sample to show the source of the data and reference this within your binned sample. Registering an environmental sample follows the same method as regular sample registration. Make sure to use the most appropriate environmental checklist and an environmental taxon (e.g. aquatic metagenome (tax id: 1169740)). Please also see here for details on how to release these samples. If no data is associated with the environmental sample, it needs to be released manually in order to be available to the public.

The methods for submitting metagenomic samples follow the same process as any other sample submission. Follow the links for more information.

• Register a Sample

Stage 2: Prepare the files

The set of files that are part of the submission are specified using a manifest file. The manifest file is specified using the -manifest <filename> option.

A binned metagenome assembly submission consists of the following files:

• 1 manifest file
• 1 fasta file

**Manifest file**

The manifest file has two columns separated by a tab (or any whitespace characters):

• Field name (first column): case insensitive field name
• Field value (second column): field value

The following metadata fields are supported in the manifest file:

• STUDY: Study accession or unique name (alias)
• SAMPLE: Binned sample accession or unique name (alias)
• ASSEMBLYNAME: Unique assembly name
• ASSEMBLY_TYPE: ‘binned metagenome’
• COVERAGE: The estimated depth of sequencing coverage
• PROGRAM: The assembly program
• PLATFORM: The sequencing platform, or comma-separated list of platforms
• MINGAPLENGTH: Minimum length of consecutive Ns to be considered a gap (optional)
• MOLECULETYPE: ‘genomic DNA’, ‘genomic RNA’ or ‘viral cRNA’ (optional)
• DESCRIPTION: Free text description of the genome assembly (optional)
• RUN_REF: Comma separated list of run accession(s) (optional)

Please see further below for validation rules affecting some of these fields.

The following file name fields are supported in the manifest file:

• FASTA: sequences in fasta format

For example, the following manifest file represents a binned metagenome assembly consisting of contigs provided in one fasta file:

```plaintext
STUDY TODO
SAMPLE TODO
ASSEMBLYNAME TODO
ASSEMBLY_TYPE TODO
COVERAGE TODO
PROGRAM TODO
PLATFORM TODO
MINGAPLENGTH TODO
MOLECULETYPE genomic DNA
FASTA  binned_metagenome.fasta.gz
```

**Stage 3: Validate and submit the files**

Files are validated, uploaded and submitted using the [Webin command line submission interface](#).

Please refer to the [Webin command line submission interface documentation](#) for more information about the submission process.
Assigned accession numbers

Once the genome assembly has been submitted an analysis (ERZxxxxxx) accession number is immediately assigned and returned to the submitter by the Webin command line submission interface.

The purpose of the ERZ accession number is for the submitter to be able to refer to their submission within the Webin submission service and access their data in the browser.

For binned metagenome assemblies, long term stable accession numbers that can be used in publications are:

- Study accession (PRJEBxxxx) assigned at time of study registration.
- Sample accession (SAMEAxxxxx) assigned at time of sample registration.

See an example of a publicly available binned metagenome at: https://www.ebi.ac.uk/ena/browser/view/ERZ1100281

Validation rules

Assembly name validation

Assembly names must: - match the pattern: ^[A-Za-z0-9][A-Za-z0-9 _#-.*]$ - not be longer than 50 characters - not include the taxonomic name of the organism assembled

Sequence validation


Submitting A Metagenome-Assembled Genome (MAG)

- What is considered a MAG in ENA?
- Introduction
- Stage 1: Pre-register study, samples and lower level assemblies
- Stage 2: Prepare the files
- Stage 3: Validate and submit the files
- Assigned accession numbers
- Validation rules

What is considered a MAG in ENA?

A Metagenome-Assembled Genome (MAG) is a single-taxon assembly based on one or more binned metagenomes that has been asserted to be a close representation to an actual individual genome (that could match an already existing isolate or represent a novel isolate).

MAG submissions are submitted at the same level as isolate genomes and are distributed within INSDC in the same way. As an environmental sample can contain many duplicate genomes of the same organism and as MAG assemblies are more prone to contamination, we request only the highest quality unique-taxon submissions are submitted as MAGs.
There should only be one MAG submitted for each species within a biome. This can be determined using a de-replication step or by choosing the highest quality representative genome for each predicted species.

It is recommended that for all MAG submissions, all lower level assemblies are submitted first. This means a MAG submission can be used to highlight the best and most representative derived assemblies from a binned metagenome set and ensures the methods used to derive your MAG are reproducible.

- Submitting A Primary Metagenome Assembly
- Submitting Binned Metagenome Assemblies

If your study is small-scale and you only intend to submit a very small number of MAGs, lower level metagenome assembly submissions may not be necessary, though are still recommended where possible. Please contact our helpdesk if you have any questions.

Introduction

Metagenome assemblies can be submitted to the European Nucleotide Archive (ENA) using the Webin command line submission interface with -context genome.

Please contact our helpdesk if you intend to submit an assembly assembled from third party data.

Each MAG from an environmental source requires a virtual derived MAG sample so please follow instructions carefully.

Genome assembly submissions include plasmids, organelles, complete virus genomes, viral segments/replicons, bacteriophages, prokaryotic and eukaryotic genomes.

A Metagenome-Assembled Genome consists of:

- General assembly information
  - Study accession or unique name (alias)
  - MAG Sample accession or unique name (alias)
  - Assembly name
  - Assembly program
  - Sequencing platform
  - Minimum gap length
  - Molecule type (genomic DNA, genomic RNA or viral cRNA)
  - Coverage
  - Free text description of the assembly (optional)
- Contig sequences (if any)
- Scaffold sequences (if any)
- Chromosome sequences (if any)
- Unlocalised sequences (if any)
- Functional annotation (optional)

For assembly submission purposes, the term ‘chromosome’ should be understood to include organelles (e.g. mitochondria and chloroplasts), plasmids and viral segments.

The below image illustrates the stages of the MAG submission process:
Stage 1: Pre-register study, samples and lower level assemblies

Each submission must be associated with a pre-registered study and a MAG sample. Genome assemblies except primary metagenomes are uniquely associated with a study and a sample. If you have not done so already, please register a study. If you intend to submit your MAG with annotation, make sure to register locus tag prefixes during this stage.

- Register a Study

It is recommended to also submit all lower level metagenomic assemblies and raw reads before submitting MAGs to help record your methods and make your data reproducible.

- Submitting Raw Reads
- Submitting A Primary Metagenome Assembly
- Submitting Binned Metagenome Assemblies

Registering MAG samples

Each MAG assembly submission must be associated with a MAG sample. This is because a MAG is not an assembly of the whole set of raw data but an assembly derived from a smaller subset of those data. These virtual samples represent the subset of that data and hold all metadata related to the taxonomy of that subset as well as methods used to derive it.
MAG samples should be as specific in taxonomy as they can be and use the specific GSC MIMAGS checklist.

Please make sure these MAG samples correctly reference the sample they were derived from (e.g. a binned sample used to submit a metagenomic bin or the environmental sample used to submit the raw reads the MAG was derived from).

This can be done from within the checklist using the mandatory “sample derived from” attribute. If the assembly was derived from multiple samples or runs you can list these with a comma separated list or range.

You should also reference the source sample in the description:

“This sample represents a MAG derived from the metagenomic sample ERSXXXXX”

**What if I only want to submit MAGs and not lower level assemblies or reads?**

If you do not intend to submit raw reads or lower level assemblies as part of your metagenome assembly submission, please still register an environmental sample and reference this within your MAG sample to show the source of the data and reference this within your MAG sample. Registering an environmental sample follows the same method as regular sample registration. Make sure to use the most appropriate environmental checklist and an environmental taxon (e.g. aquatic metagenome (tax id: 1169740)). Please also see here for details on how to release these samples. If no data is associated with the environmental sample, it needs to be released manually in order to be available to the public.

The methods for submitting metagenomic samples follow the same process as any other sample submission. Follow the links for more information.

- Register a Sample

**Stage 2: Prepare the files**

The set of files that are part of the submission are specified using a manifest file. The manifest file is specified using the `-manifest <filename>` option.

The files required for submission of a genome assembly depends on the assembly level:

- Contig assembly
- Scaffold assembly
• Chromosome assembly

Contig assembly

Consists of the following files:
  • 1 manifest file
  • 1 FASTA file OR 1 flat file
This assembly level only requires information on the sequences and annotation (if any). You will receive an error if less than 2 or more than 1,000,000 sequences are submitted. If you have less than 2 sequences, then you will need to submit at a higher assembly level or as template sequences.

Scaffold assembly

Consists of the following files:
  • 1 manifest file
  • 1 FASTA file OR 1 flat file
  • 1 AGP files
This assembly level requires information on the sequences and annotation (if any). It also allows the submitter to provide an AGP file to give instructions for the assembly of the scaffolds from the contigs.

Chromosome assembly

Consists of the following files:
  • 1 manifest file
  • 1 FASTA file OR 1 flat file
  • 1 chromosome list file
  • 0-1 unlocalised list files
  • 0-1 AGP files
This assembly level allows the submission of fully assembled chromosomes (including organelles, plasmids, and viral segments). This requires information on the sequences and annotation (if any) and submission of a chromosome list file to indicate which sequences represent which ‘chromosomes’.

If these chromosomes contain unlocalised sequences (where the chromosome of the sequence is known but not the exact location) you can submit an additional unlocalised list file. However, please note, if you wish to submit unplaced contigs, you will have to submit at a lower level and use an AGP file to indicate which scaffolds/contigs are assembled to form each chromosome. Any sequences that are not used to assemble chromosomes are considered unplaced.

For this assembly level in particular, it is important to understand how sequence names are formatted so they can be consistent between files otherwise the system will just register your submission at contig level.

Sequence names

Sequences must have a unique name within the submission that is provided in the fasta, AGP or flat files. It is essential that the sequence names are unique and used consistently between files.
For example, the chromosome list file must refer to the chromosome sequences using the unique sequence names. Similarly, an AGP file must refer to scaffolds or contigs using the unique sequence names.

**Manifest file**

The manifest file has two columns separated by a tab (or any whitespace characters):

- Field name (first column): case insensitive field name
- Field value (second column): field value

The following metadata fields are supported in the manifest file:

- **STUDY**: Study accession or unique name (alias)
- **SAMPLE**: MAG sample accession or unique name (alias)
- **ASSEMBLYNAME**: Unique assembly name
- **ASSEMBLY_TYPE**: ‘Metagenome-Assembled Genome (MAG)’
- **COVERAGE**: The estimated depth of sequencing coverage
- **PROGRAM**: The assembly program
- **PLATFORM**: The sequencing platform, or comma-separated list of platforms
- **MINGAPLENGTH**: Minimum length of consecutive Ns to be considered a gap (optional)
- **MOLECULETYPE**: ‘genomic DNA’, ‘genomic RNA’ or ‘viral cRNA’ (optional)
- **DESCRIPTION**: Free text description of the genome assembly (optional)
- **RUN_REF**: Comma separated list of run accession(s) (optional)

Please see further below for validation rules affecting some of these fields.

The following file name fields are supported in the manifest file:

- **FASTA**: sequences in fasta format
- **FLATFILE**: sequences in EMBL-Bank flat file format
- **AGP**: sequences in AGP format
- **CHROMOSOME_LIST**: list of chromosomes

For example, the following manifest file represents a MAG consisting of contigs provided in one FASTA file:

```
STUDY TODO
SAMPLE TODO
ASSEMBLYNAME TODO
ASSEMBLY_TYPE TODO
COVERAGE TODO
PROGRAM TODO
PLATFORM TODO
MINGAPLENGTH TODO
MOLECULETYPE genomic DNA
FASTA metagenome_assembled_genome.fasta.gz
```
Stage 3: Validate and submit the files

Files are validated, uploaded and submitted using the Webin command line submission interface. Please refer to the Webin command line submission interface documentation for more information about the submission process.

Assigned accession numbers

Once the genome assembly has been submitted an analysis (ERZxxxxxx) accession number is immediately assigned and returned to the submitter by the Webin command line submission interface (Webin-CLI).

ERZ accessions should not be used to reference the assembly in publications. The purpose of the ERZ accession number is for the submitter to be able to refer to their submission within the Webin submission service. For example, the submitter can retrieve the assigned genome assembly and sequence accessions from the Webin submissions portal or from the Webin reports service using the ERZ accession number. This accession should be used to refer to the assembly in any conversations with helpdesk staff.

For metagenome assemblies, long term stable accession numbers that can be used in publications are:

- Study accession (PRJEBxxxxxx) assigned at time of study registration.
- Sample accession (SAMEAxxxxxx) assigned at time of sample registration.
- Genome assembly accession (GCA_xxxxxxx) assigned once the assembly has been fully processed by ENA.
- Sequence accession(s) assigned once the assembly has been fully processed by ENA.

Submitters can retrieve the genome and sequence accession numbers from the Webin submissions portal or from the Webin reports service. These accession numbers are also sent to the submitters by e-mail.

Validation rules

Assembly submissions are subject to a great deal of validation before submission is allowed. Some key points are described here.

Sample And Study Validation

- Sample and study (BioProject) pair must be unique for an assembly (except primary metagenomes)
- Sample taxonomic classification must be species rank or below (or equivalent) within NCBI taxonomy.

Assembly name validation

Assembly names must:

- match the pattern: ^[A-Za-z0-9][A-Za-z0-9 _#-.*]*$
- not be longer than 50 characters
- not include the name of the organism assembled
Chromosome name validation

Chromosome names must:

- match the pattern: `^[A-Za-z0-9][A-Za-z0-9_\#-_.]*$`
- be shorter than 33 characters
- not contain any of the following as part of their name (case insensitive):
  - ‘chr’
  - ‘chrn’
  - ‘chrom’
  - ‘chromosome’
  - ‘linkage group’
  - ‘linkage-group’
  - ‘linkage_group’
  - ‘plasmid’
- be unique within an assembly

Sequence validation

Sequences must:

- have unique names within an assembly
- be at least 20bp long
- not have terminal Ns

Note: Submissions of binned metagenomes or Metagenome-Assembled Genomes follow different rules as they are considered derived assemblies.

When it comes to submission of derived assemblies you will be asked to register additional ‘virtual’ samples for each assembly (a binned sample or a MAG sample).
This is because a binned assembly or MAG is not an assembly of the whole set of raw metagenomic data but in fact an assembly derived from a smaller subset of those data. These virtual samples represents the subset of that data and hold all metadata related to the taxonomy of that subset as well as methods used to derive it.

All derived assemblies from an environmental source require a virtual derived sample so please follow instructions carefully.

### 6.5.3 Submitting Environmental Single-Cell Amplified Genomes

- **Introduction**
- **Stage 1: Pre-register study and samples**
- **Stage 2: Prepare the files**
- **Stage 3: Validate and submit the files**
- **Assigned accession numbers**
- **Validation rules**

**Introduction**

Environmental SAG assemblies can be submitted to the European Nucleotide Archive (ENA) using the Webin command line submission interface with `-context genome`.

Please contact our helpdesk if you intend to submit an assembly assembled from third party data.

Each SAG from an environmental source requires a virtual derived SAG sample so please follow instructions carefully.

Genome assembly submissions include plasmids, organelles, complete virus genomes, viral segments/replicons, bacteriophages, prokaryotic and eukaryotic genomes.

An environmental SAG assembly consists of:

- General assembly information
  - Study accession or unique name (alias)
  - **Environmental** Sample accession or unique name (alias)
  - **SAG** Sample accession or unique name (alias)
Stage 1: Pre-register study and samples

Each submission must be associated with a pre-registered study and a SAG sample.

Genome assemblies except primary metagenomes are uniquely associated with a study and a sample.

If you have not done so already, please register a study. If you intend to submit your SAG with annotation, make sure to register locus tag prefixes during this stage.

- Register a Study
It is also strongly recommended to submit the reads from which the assembly was assembled. When submitting the raw reads in a Single-Cell Amplification study, these should remain multiplexed in BAM format and be submitted to an environmental sample. This sample should use the most appropriate environmental checklist and an environmental taxon (e.g. aquatic metagenome (tax id: 1169740)).

- Submitting Raw Reads

If you do not intend to submit raw reads as part of your SAG submission, please make sure you have registered environmental samples anyway. These will represent the original sequenced biomaterial that your SAG was derived from. You will also need to follow these guidelines for details on how to release your environmental samples. If no data is associated with a sample, it needs to be released manually in order to be available to the public.

**SAG sample registration**

Each SAG assembly submission must be associated with a SAG sample. This is because a SAG is not an assembly of the whole set of raw data but an assembly derived from a smaller subset of those data. These virtual samples represent the subset of that data and hold all metadata related to the taxonomy of that subset as well as methods used to derive it.

It should be as specific in taxonomy as it can be and use the specific GSC MISAGS checklist. Please make sure these SAG samples correctly reference the environmental sample that the SAG was derived from. This can be done from within the checklist using the mandatory “sample derived from” attribute. If the assembly was derived from multiple samples or runs you can list these with a comma separated list or range.

You should also reference the source sample in the description:

“This sample represents a Single-Cell Amplified Genome derived from the environmental sample ERSXXXXX”

- Register a Sample

**Stage 2: Prepare the files**

The set of files that are part of the submission are specified using a manifest file. The manifest file is specified using the `-manifest <filename>` option.

The files required for submission of a genome assembly depends on the assembly level:

- *Contig assembly*
• Scaffold assembly
• Chromosome assembly

Contig assembly

Consists of the following files:

• 1 manifest file
• 1 FASTA file OR 1 flat file

This assembly level only requires information on the sequences and annotation (if any). You will receive an error if less than 2 or more than 1,000,000 sequences are submitted. If you have less than 2 sequences, then you will need to submit at a higher assembly level or as template sequences.

Scaffold assembly

Consists of the following files:

• 1 manifest file
• 1 FASTA file OR 1 flat file
• 1 AGP files

This assembly level requires information on the sequences and annotation (if any). It also allows the submitter to provide an AGP file to give instructions for the assembly of the scaffolds from the contigs.

Chromosome assembly

Consists of the following files:

• 1 manifest file
• 1 FASTA file OR 1 flat file
• 1 chromosome list file
• 0-1 unlocalised list files
• 0-1 AGP files

This assembly level allows the submission of fully assembled chromosomes (including organelles, plasmids, and viral segments). This requires information on the sequences and annotation (if any) and submission of a chromosome list file to indicate which sequences represent which ‘chromosomes’.

If these chromosomes contain unlocalised sequences (where the chromosome of the sequence is known but not the exact location) you can submit an additional unlocalised list file. However, please note, if you wish to submit unplaced contigs, you will have to submit at a lower level and use an AGP file to indicate which scaffolds/contigs are assembled to form each chromosome. Any sequences that are not used to assemble chromosomes are considered unplaced.

For this assembly level in particular, it is important to understand how sequence names are formatted so they can be consistent between files otherwise the system will just register your submission at contig level.
Sequence names

Sequences must have a unique name within the submission that is provided in the fasta, AGP or flat files. It is essential that the sequence names are unique and used consistently between files.

For example, the chromosome list file must refer to the chromosome sequences using the unique sequence names. Similarly, an AGP file must refer to scaffolds or contigs using the unique sequence names.

Manifest file

The manifest file has two columns separated by a tab (or any whitespace characters): - Field name (first column): case insensitive field name - Field value (second column): field value

The following metadata fields are supported in the manifest file:

- STUDY: Study accession or unique name (alias)
- SAMPLE: SAG Sample accession or unique name (alias)
- ASSEMBLYNAME: Unique assembly name
- ASSEMBLY_TYPE: ‘Environmental Single-Cell Amplified Genome (SAG)’
- COVERAGE: The estimated depth of sequencing coverage
- PROGRAM: The assembly program
- PLATFORM: The sequencing platform, or comma-separated list of platforms
- MINGAPLENGTH: Minimum length of consecutive Ns to be considered a gap (optional)
- MOLECULETYPE: ‘genomic DNA’, ‘genomic RNA’ or ‘viral cRNA’ (optional)
- DESCRIPTION: Free text description of the genome assembly (optional)
- RUN_REF: Comma separated list of run accession(s) (optional)

Please see further below for validation rules affecting some of these fields.

The following file name fields are supported in the manifest file:

- FASTA: sequences in fasta format
- FLATFILE: sequences in EMBL-Bank flat file format
- AGP: Sequences in AGP format
- CHROMOSOME_LIST: list of chromosomes

For example, the following manifest file represents an environmental single-cell assembly consisting of contigs provided in one FASTA file:

```
STUDY    TODO
SAMPLE   TODO
ASSEMBLYNAME  TODO
ASSEMBLY_TYPE  Environmental Single-Cell Amplified Genome (SAG)
COVERAGE   TODO
PROGRAM    TODO
PLATFORM   TODO
MINGAPLENGTH TODO
MOLECULETYPE  genomic DNA
FASTA     single-cell Genome.fasta.gz
```

6.5. Submission Options 133
Stage 3: Validate and submit the files

Files are validated, uploaded and submitted using the Webin command line submission interface. Please refer to the Webin command line submission interface documentation for more information about the submission process.

Assigned accession numbers

Once the genome assembly has been submitted an analysis (ERZxxxxxxxx) accession number is immediately assigned and returned to the submitter by the Webin command line submission interface (Webin-CLI).

ERZ accessions should not be used to reference the assembly in publications. The purpose of the ERZ accession number is for the submitter to be able to refer to their submission within the Webin submission service. For example, the submitter can retrieve the assigned genome assembly and sequence accessions from the Webin submissions portal or from the Webin reports service using the ERZ accession number. This accession should be used to refer to the assembly in any conversations with helpdesk staff.

For Environmental Single-cell Amplified Genome assemblies, long term stable accession numbers that can be used in publications are:

- Study accession (PRJEBxxxxxx) assigned at time of study registration.
- Sample accession (SAMEAxxxxxx) assigned at time of sample registration.
- Genome assembly accession (GCA_xxxxxxxx) assigned once the assembly has been fully processed by ENA.
- Sequence accession(s) assigned once the assembly has been fully processed by ENA.

Submitters can retrieve the genome and sequence accession numbers from the Webin submissions portal or from the Webin reports service. These accession numbers are also sent to the submitters by e-mail.

Validation rules

Assembly submissions are subject to a great deal of validation before submission is allowed. Some key points are described here.

Sample And Study Validation

- Sample and study (BioProject) pair must be unique for an assembly (except primary metagenomes)
- Sample taxonomic classification must be species rank or below (or equivalent) within NCBI taxonomy.

Assembly name validation

Assembly names must:

- match the pattern: ^[A-Za-z0-9][A-Za-z0-9 _#-.]*$
- not be longer than 50 characters
- not include the name of the organism assembled
Chromosome name validation

Chromosome names must:

- match the pattern: `^[A-Za-z0-9][A-Za-z0-9_#-\.]*$`
- be shorter than 33 characters
- not contain any of the following as part of their name (case insensitive):
  - `chr`
  - `chrm`
  - `chrom`
  - `chromosome`
  - `linkage group`
  - `linkage-group`
  - `linkage_group`
  - `plasmid`
- be unique within an assembly

Sequence validation

Sequences must:
- have unique names within an assembly
- be at least 20bp long
- not have terminal Ns
- consist of bases: `'a','c','g','t','u','b','d','h','k','m','n','r','s','v','w','y'`

6.5.4 Submitting Transcriptome Assemblies

- Introduction
- Stage 1: Pre-register study and sample
- Stage 2: Prepare the files
- Stage 3: Validate and submit the files
- Assigned accession numbers
- Validation rules

Introduction

Transcriptome assemblies can be submitted to the European Nucleotide Archive (ENA) using the Webin command line submission interface with `-context transcriptome` option.

A transcriptome assembly consists of:

- General assembly information
  - Study accession or unique name (alias)
  - Sample accession or unique name (alias)
  - Assembly name
  - Assembly program

6.5. Submission Options 135
Stage 1: Pre-register study and sample

Each submission must be associated with a pre-registered study and a sample. For transcriptomic assemblies, raw reads must also be submitted to give context to the data.

- Register a Study
- Register a Sample
- Submit Read Data

Stage 2: Prepare the files

The set of files that are part of the submission are specified using a manifest file. The manifest file is specified using the `--manifest <filename>` option.

A transcriptome assembly submission consists of the following files:

- 1 manifest file
- 1 FASTA file OR 1 flat file

Transcriptome assemblies should be of a high enough quality to fulfil the following criteria:

1. They must have at least 1x coverage by primary sequence at each base. Regions of a TSA record can be assembled from a single Expressed Sequence Tag (EST) or read so that coverage is only 1x.
2. Bases that are listed as ‘n’ should be less than 10%.
3. They should not have a stretch of more than 15 n’s in a row. If they are within 20 base pairs of the c- or n-terminus they should be removed.
4. No assemblies can be shorter than 200 base pairs.
5. Any vector sequence or primers should be removed.

**Manifest file**

The manifest file has two columns separated by a tab (or any whitespace characters):

- Field name (first column): case insensitive field name
- Field value (second column): field value

The following metadata fields are supported in the manifest file:

- **STUDY**: Study accession or unique name (alias)
- **SAMPLE**: Sample accession or unique name (alias)
- **ASSEMBLYNAME**: The unique assembly name
- **ASSEMBLY_TYPE**: ‘isolate’
- **PROGRAM**: The assembly program
- **PLATFORM**: The sequencing platform, or comma-separated list of platforms
- **DESCRIPTION**: Free text description of the transcriptome assembly (optional)
- **RUN_REF**: Comma separated list of run accession(s) (optional)

The following file name fields are supported in the manifest file:

- **FASTA**: sequences in FASTA format
- **FLATFILE**: sequences in EMLB-Bank flat file format

For example, the following manifest file represents a genome assembly consisting of transcripts provided in one FASTA file:

```
STUDY  TODO
SAMPLE TODO
ASSEMBLYNAME  TODO
ASSEMBLY_TYPE  isolate
PROGRAM TODO
PLATFORM TODO
FASTA  transcriptome.fasta.gz
```

**Fasta file**

Unannotated sequences should be submitted as a Fasta file.

The sequence name is extracted from the fasta header. For example the following header contains the name ‘contig1’:

```
>contig1
```

**Flat file**

Annotated sequences must be submitted using an EMLB-Bank flat file.

The sequence name is extracted from the AC * line and must be prefixed with a ‘_’. For example the following AC * line defines name ‘contig1’:
Stage 3: Validate and submit the files

Files are validated, uploaded and submitted using the Webin command line submission interface (Webin-CLI). Please refer to the Webin command line submission interface documentation for more information about the submission process.

Assigned accession numbers

Once the genome assembly has been submitted an analysis (ERZxxxxxx) accession number is immediately assigned and returned to the submitter by the Webin command line submission interface (Webin-CLI).

ERZ accessions should not be used to reference the assembly in publications. The purpose of the ERZ accession number is for the submitter to be able to refer to their submission within the Webin submission service. For example, the submitter can retrieve the assigned sequence accessions from the Webin submissions portal or from the Webin reports service using the ERZ accession number. Sequence accessions are sent to the user by email once the TSA has completed processing. The ERZ accession should be used to refer to the assembly in any conversations with helpdesk staff.

For transcriptome assemblies, long term stable accession numbers that can be used in publications are:

- Study accession (PRJEBxxxxxx) assigned at time of study registration
- Sample accession (SAMEAxxxxxx) assigned at time of study registration
- Sequence accession(s) assigned once the genome assembly submission has been fully processed by ENA

See an example of a publicly available TSA at: https://www.ebi.ac.uk/ena/browser/view/GAAJ01000000

Validation rules

Sequence validation rules

Sequences must:

- have unique names within an assembly
- be at least 200bp long
- not have terminal Ns

6.5.5 Submitting Metatranscriptome Assemblies

- Introduction
- Stage 1: Pre-register study and sample
- Stage 2: Prepare the files
- Stage 3: Validate and submit the files
- Assigned accession numbers
- Validation rules
Introduction

Metatranscriptome assemblies can be submitted to the European Nucleotide Archive (ENA) using the Webin command line submission interface with `-context transcriptome` option.

A metatranscriptome assembly consists of:

- General assembly information
  - Study accession or unique name (alias)
  - Environmental Sample accession or unique name (alias)
  - Assembly program
  - Sequencing platform
- Sequences
- Functional annotation (optional)

The following picture illustrates the stages of the transcriptome assembly submission process:

**Stage 1: Pre-register study and sample**

Each submission must be associated with a pre-registered study and a pre-registered environmental sample. This should be the same sample used for submitting raw reads. Please make sure the appropriate environmental checklist is chosen for this and an environmental taxon is used (e.g. aquatic metagenome (tax id: 1169740)). See the available environmental taxa in the [ENA Tax Portal](https://www.ebi.ac.uk/ena/browser/view/Taxon:408169). Click on the Tax tree tab and click the ‘+’ icons to expand the categories.

For transcriptomic assemblies, raw reads must also be submitted to give context to the data.

The methods for submitting these studies follow the same process as any other study/sample/read submission. Follow the links for more information.

Instructions for interactive submitters:

- Register a Study
- Register a Sample

6.5. Submission Options
Stage 2: Prepare the files

The set of files that are part of the submission are specified using a manifest file. The manifest file is specified using the \(-manifest \text{ <filename>}\) option.

A transcriptome assembly submission consists of the following files:

- 1 manifest file
- 1 FASTA file OR 1 flat file

Transcriptome assemblies should be of a high enough quality to fulfil the following criteria:

1. They must have at least 1x coverage by primary sequence at each base. Regions of a TSA record can be assembled from a single Expressed Sequence Tag (EST) or read so that coverage is only 1x.
2. Bases that are listed as ‘n’ should be less than 10%.
3. They should not have a stretch of more than 15 n’s in a row. If they are within 20 base pairs of the c- or n-terminus they should be removed.
4. No assemblies can be shorter than 200 base pairs.
5. Any vector sequence or primers should be removed.

Manifest file

The manifest file has two columns separated by a tab (or any whitespace characters): - Field name (first column): case insensitive field name - Field value (second column): field value

The following metadata fields are supported in the manifest file:

- **STUDY**: Study accession or unique name (alias)
- **SAMPLE**: Sample accession or unique name (alias)
- **ASSEMBLYNAME**: The unique assembly name
- **ASSEMBLY_TYPE**: ‘metatranscriptome’ (only valid for Webin-CLI v3.0.0 or later)
- **PROGRAM**: The assembly program
- **PLATFORM**: The sequencing platform, or comma-separated list of platforms
- **RUN_REF**: Comma separated list of run accession(s) (optional)

The following file name fields are supported in the manifest file:

- **FASTA**: sequences in fasta format
- **FLATFILE**: sequences in EMBL-Bank flat file format

For example, the following manifest file represents a metatranscriptome assembly provided in one fasta file:

<table>
<thead>
<tr>
<th>STUDY</th>
<th>TODO</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAMPLE</td>
<td>TODO</td>
</tr>
<tr>
<td>ASSEMBLYNAME</td>
<td>TODO</td>
</tr>
<tr>
<td>ASSEMBLY_TYPE</td>
<td>metatranscriptome</td>
</tr>
<tr>
<td>PROGRAM</td>
<td>TODO</td>
</tr>
<tr>
<td>PLATFORM</td>
<td>TODO</td>
</tr>
<tr>
<td>FASTA</td>
<td>metatranscriptome.fasta.gz</td>
</tr>
</tbody>
</table>
Fasta file

Unannotated sequences should be submitted as a Fasta file.

The sequence name is extracted from the fasta header. For example the following header contains the name ‘contig1’:

>contig1

Flat file

Annotated sequences must be submitted using an EMBL-Bank flat file.

The sequence name is extracted from the AC * line and must be prefixed with a ‘_’. For example the following AC * line defines name ‘contig1’:

AC * _contig1

Stage 3: Validate and submit the files

Files are validated, uploaded and submitted using the Webin command line submission interface (Webin-CLI). Please refer to the Webin command line submission interface documentation for more information about the submission process.

Assigned accession numbers

Once the genome assembly has been submitted an analysis (ERZxxxxxx) accession number is immediately assigned and returned to the submitter by the Webin command line submission interface (Webin-CLI).

ERZ accessions should not be used to reference the assembly in publications. The purpose of the ERZ accession number is for the submitter to be able to refer to their submission within the Webin submission service. For example, the submitter can retrieve the assigned genome assembly and sequence accessions from the Webin submissions portal or from the Webin reports service using the ERZ accession number. This accession should be used to refer to the assembly in any conversations with helpdesk staff.

For transcriptome assemblies, long term stable accession numbers that can be used in publications are:

- Study accession (PRJEBxxxxxx) assigned at time of study registration
- Sample accession (SAMEAxxxxxx) assigned at time of study registration
- Sequence accession(s) assigned once the genome assembly submission has been fully processed by ENA

Submitters can retrieve the genome and sequence accession numbers from the Webin submissions portal or from the Webin reports service. These accession numbers are also sent to the submitters by e-mail.

See an example of a publicly available metatranscriptome TSA at: https://www.ebi.ac.uk/ena/browser/view/HAZG01000000

Validation rules

Sequence validation rules

Sequences must:

- have unique names within an assembly
- be at least 200bp long
• not have terminal Ns
7.1 Introduction

The information on this page pertains to the submission of targeted sequences, typically short assembled and annotated sequences representing interesting features or gene regions. This is unrelated to the submission of full chromosome or genome assemblies, which is described in our Assembly Submission Pages.

All submissions of this type are submitted as ‘analysis’ objects, but the accessions of these analyses are not exposed. Instead, specific sequence accessions are assigned later, and these are what should be used to reference the sequences. Sequence submission requires that a study be registered first. Please do this before attempting any of the guides shown below:

• Register a Study
Note: This submission route is for sets of stand-alone targeted assembled and annotated sequences only. If you intend to submit an annotated assembly such as a genome, please follow the assembly submission guidelines and submit your assembly in EMBL flat file format.

7.2 Accessions

As all sequences in ENA are submitted as ‘analyses’, for each sequence set submission, Webin will report a unique accession number that starts with ERZ. This accession number is for internal processing only and will not be visible in the browser. Furthermore, if your submission includes multiple sequences, the ERZ accession will not be unique to any one of them. Instead, you will receive additional post-processing accession numbers for your sequences via email, each of which will be unique to one of your sequences.

Always make a note of any accessions you receive as these are the unique identifiers for each of your submissions to ENA.

7.3 Submission Options

Many types of sequence can be submitted using a checklist. These define a set of metadata which should be provided for a given type of sequence and streamline the submission process for these sequence types. Checklist submission allows you to avoid having to create the flatfile record manually.

Therefore, please check the list of available checklists to determine whether one of them meets the needs of your submission:

• List of Annotation Checklists

7.3.1 If There Is A Suitable Checklist Available

You can use the Webin Interactive service to submit your checklist sequences through web forms. You also have the option of preparing a spreadsheet and submitting it through our Webin-CLI tool.

Submit Annotated Sequences Interactively

Submitting a new annotated sequence entry can be done interactively through Webin. Ensure you register a study before you begin if you have not already done so.

Interactive submission of annotated sequences requires you to select a checklist and fill out a spreadsheet of information about your sequence. Please view the available Annotation Checklists and pick out a suitable one in advance. If there is no appropriate checklist available, you will need to create a flatfile and use a different submission method.

Two methods are described here: spreadsheet based and non-spreadsheet based. Submitting via spreadsheet is recommended, especially if you have many sequences to submit.

Submitting Via Spreadsheet

To begin, visit the sequence checklist portal.
Here, you can review the available checklists, select the most appropriate, customise it, and review the allowed values for each field. Download your chosen spreadsheet and fill it out according to the restrictions given in the checklist portal. You should not edit any of the filled cells of the spreadsheet you download, only modify the empty ones by inserting your data and metadata.

To submit, log in to your Webin account, navigate to the ‘New Submission’ tab and select ‘Submit other assembled and annotated sequences’.

On the next screen, select the study you wish to submit to (or use the ‘Create a new study’ link).

On the following screen, note the ‘Submit Completed Spreadsheet’ button. You should use this to upload your selected checklist. If there are errors with your spreadsheet, you will be informed: correct them and redo the submission.

Once the submission is accepted, you will be shown a confirmation screen with a tracking number resembling ERZxxxxxx. You will later receive an email with your accession numbers.

Non-Spreadsheet Submission

Submitting without a spreadsheet can be convenient if you have a small number of sequences (<3) to submit.

Step 1: Select and Customise a Checklist

To begin, log in to your Webin account, navigate to the ‘New Submission’ tab and select ‘Submit other assembled and annotated sequences’.

On the next screen, select the study you wish to submit to (or use the ‘Create a new study’ link).

Next, you can start building your submission by clicking ‘Select Checklist’.

Browse the list of available checklists. A description is available for each one.
You can select extra fields to add to your spreadsheet at this point. These extra fields give your data more context and help other users understand it.
Step 2: Fill Out Shared Details

In this next screen, you should fill out details which are shared by all of your sequences. This information will be used to populate the form for each sequence you add to this submission. None of these fields are mandatory at this stage, and you will be able to edit their content on an individual basis later.

Step 3: Fill Out Individual Details

In this final step, you should fill out the details for each sequence individually. Depending on how you completed the previous step, some fields will be filled already.
Use the ‘+Add’ button to increase the amount of sequences, using the adjacent field to specify how many sequences to add. In the box below, you can move between different sequences using the checks on the left. On the right, a green tick indicates a sequence with all mandatory fields completed, while a red exclamation mark indicates errors to be resolved. The red ‘X’ can be used to remove an entry completely.

Next to each field, you can mouse over the blue ‘i’ to learn more about what should be entered.

### Step 4: Submit Your Sequences

When you have entered all the information you wish to and all of the sequences show green ticks indicating that they are correctly filled out, click ‘Submit’ to have your submission validated. If your sequences fail validation, you will be informed of this here. Otherwise, if your submission is successful, a confirmation page will appear and you will later receive an email with your accession numbers.

### Submit Annotated Sequence Spreadsheets with Webin-CLI

#### Introduction

Annotated sequences (e.g. 16S rRNA genes) can be submitted to the European Nucleotide Archive (ENA) as tab-separated (tsv) spreadsheets using the Webin command line submission interface with \(-context\) sequence option.

An annotated sequence submission consists of:
• General sequence information
  – Study accession or unique name (alias)
  – Unique name for the submission
  – Free text description of the set of submitted sequences (optional)
• Sequences
• Functional annotation

The following picture illustrates the stages of the annotated sequence spreadsheet submission process:

**Stage 1: Pre-register study**

Each submission must be associated with a pre-registered study.
  • Register a Study

**Stage 2: Prepare the files**

The set of files that are part of the submission are specified using a manifest file. The manifest file is specified using the `-manifest <filename>` option.

An annotated sequence spreadsheet submission consists of the following files:
  • 1 manifest file
  • 1 tab-separated (tsv) spreadsheet containing the sequences and functional annotation
Manifest file

The manifest file has two columns separated by a tab (or any whitespace characters):

- Field name (first column): case insensitive field name
- Field value (second column): field value

The following metadata fields are supported in the manifest file:

- STUDY: Study accession or unique name (alias)
- NAME: Unique name for the submission
- DESCRIPTION: Free text description of the set of submitted sequences (optional)

The following file name fields are supported in the manifest file:

- TAB: tab-separated (tsv) spreadsheet containing the sequences and functional annotation

For example, the following manifest file represents a submission:

<table>
<thead>
<tr>
<th>STUDY</th>
<th>TODO</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>TODO</td>
</tr>
<tr>
<td>TAB</td>
<td>sequences.tsv.gz</td>
</tr>
</tbody>
</table>

Tab-separated (tsv) spreadsheet

Please download and fill a tab-separated (tsv) spreadsheet template from the Webin submission portal:

https://www.ebi.ac.uk/ena/submit/webin

- Step 1: Expand the ‘Download spreadsheet template for annotated sequences’ option from the ‘Submit’ page.
- Step 2: Press the ‘Start’ button.
- Step 3: Select the most appropriate checklist group.
Download spreadsheet template for annotated sequences

Please select the most appropriate checklist group, checklist and checklist fields. Download an empty spreadsheet template, Fill in the spreadsheet and submit the spreadsheet using Webin command line interface.

1. Select checklist group

Please select a checklist group first.

- Frequently-Used Checklists (e.g. rRNA gene, coding genes, mRNAs, MHC genes)
- Marker Sequence Checklists (e.g. COI, ITS, matK, D-loop, IGS)
- Virus-Specific Checklists (e.g. viral coding genes, UTR, viroids and alpha/beta-satellites)
- Large-Scale Data Checklists (e.g., EST, GSS, STS)
- All Checklists

2. Select checklist

3. Select checklist fields

4. Download spreadsheet template

- Step 4: Select the most appropriate checklist.
Select checklist group

Select checklist

You have selected **Frequently-Used Checklists (e.g. rRNA gene, coding genes, mRNAs, MHC genes).** Please select the most appropriate checklist from the list below.

**rRNA gene**

For ribosomal RNA genes from prokaryotic, nuclear or organellar DNA. All rRNAs are considered partial.

**Single CDS genomic DNA**

For complete or partial coding sequence (CDS) derived from genomic DNA. This checklist will not accept segmented genes (i.e., with intron regions) so should be used for prokaryotic, organellar genes or for submitting a single exon.

**MHC gene 1 exon**

For partial MHC class I or II antigens containing one exon ONLY.

**Gene Promoter**

For submission of uni- or bi-directional gene promoter regions.

- Step 5: Select the checklist fields and click ‘Next’ at the bottom of the page.
Select checklist group

Select checklist

Select checklist fields

You have selected **Gene Promoter**. Please select the checklist fields below.

**Fundamental information**

- **ORGANISM_NAME**
  Formal taxonomic name of organism. Example: Escherichia coli, uncultured bacterium, Nicotiana tabacum.

- **ORGANELLE**
  Mandatory if a sequence originates from an intracellular structure other than nucleus. Example: mitochondrion, chloroplast, plastid.

- **BI_PROM**
  Question to ascertain whether the promoter acts on multiple genes. If a question to ascertain whether the promoter acts on multiple genes.

• Step 6: Click ‘Download’ button to download the spreadsheet template.
Stage 3: Validate and submit the files

Files are validated, uploaded and submitted using the Webin command line submission interface. Please refer to the Webin command line submission interface documentation for more information about the submission process.

Assigned accession numbers

Once the sequences have been submitted an analysis (ERZ) accession number is immediately assigned and returned to the submitter by the Webin command line submission interface.

The purpose of the ERZ accession number is for the submitter to be able to refer to their submission within the Webin submission service. For example, the submitter can retrieve the assigned sequence accessions from the Webin submissions portal or from the Webin reports service using the ERZ accession number.
For sequences, long term stable accession numbers that can be used in publications are:

- Study accession (PRJ) assigned at time of study registration.
- Sequence accession(s) assigned once the submission has been fully processed by ENA.

Submitters can retrieve the sequence accession numbers from the Webin submissions portal or from the Webin reports service. These accession numbers are also sent to the submitters by e-mail.

### 7.3.2 If No Suitable Checklist Exists

You will need to prepare your sequence in the EMBL Flat File Format and submit it using Webin-CLI as described in the below link. Webin-CLI will validate the file for you before accepting its submission, and will advise you of any formatting errors.

**Submit Annotated Sequence Flatfiles with Webin-CLI**

**Introduction**

Annotated sequence flat files can be submitted to the European Nucleotide Archive (ENA) using the Webin command line submission interface with `-context sequence` option.

An annotated sequence submission consists of:

- General sequence information
  - Study accession or unique name (alias)
  - Unique name for the submission
  - Free text description of the submitted sequences (optional)
- Sequences in EMBL-Bank flat file format

The following picture illustrates the stages of the annotated sequence flat file submission process:

![Stages of submission process]

**Stage 1**

**REGISTER STUDY**

**Stage 2**

**PREPARE FILES**

**Stage 3**

**VALIDATE FILES** **SUBMIT FILES**

Submission
Stage 1: Pre-register study

Each submission must be associated with a pre-registered study.

- Register a study

Stage 2: Prepare the files

The set of files that are part of the submission are specified using a manifest file. The manifest file is specified using the -manifest <filename> option.

An annotated sequence flat file submission consists of the following files:

- 1 manifest file
- 1 flat file containing the sequences and functional annotation

Manifest file

The manifest file has two columns separated by a tab (or any whitespace characters):

- Field name (first column): case insensitive field name
- Field value (second column): field value

The following metadata fields are supported in the manifest file:

- STUDY: Study accession or unique name (alias)
- NAME: Unique name for the submission
- DESCRIPTION: Free text description of the submitted sequences (optional)

The following file name fields are supported in the manifest file:

- FLATFILE: sequences in EMBL-Bank flat file format

For example, the following manifest file represents a submission:

<table>
<thead>
<tr>
<th>STUDY</th>
<th>TODO</th>
</tr>
</thead>
<tbody>
<tr>
<td>NAME</td>
<td>TODO</td>
</tr>
<tr>
<td>FLATFILE</td>
<td>sequences.dat.gz</td>
</tr>
</tbody>
</table>

Stage 3: Validate and submit the files

Files are validated, uploaded and submitted using the Webin command line submission interface. Please refer to the Webin command line submission interface documentation for more information about the submission process.

Assigned accession numbers

Once the sequences have been submitted an analysis (ERZ) accession number is immediately assigned and returned to the submitter by the Webin command line submission interface.

The purpose of the ERZ accession number is for the submitter to be able to refer to their submission within the Webin submission service. For example, the submitter can retrieve the assigned sequence accessions from the Webin submissions portal or from the Webin reports service using the ERZ accession number.
For sequences, long term stable accession numbers that can be used in publications are:

- Study accession (PRJ) assigned at time of study registration.
- Sequence accession(s) assigned once the submission has been fully processed by ENA.

Submitters can retrieve the sequence accession numbers from the Webin submissions portal or from the Webin reports service. These accession numbers are also sent to the submitters by e-mail.
8.1 Introduction

Any secondary analyses you wish to submit to ENA that are not Assemblies or Annotated Sequences need to be submitted programmatically.

All submissions to ENA must be part of an ENA Study. This gives context to the data being submitted and controls the release date of the associated data.
Other submission objects may also need submitting before you can submit your secondary analyses depending on the analysis type. E.g. read alignments require a raw read submission as well as a Study.

The analysis XML format is defined by SRA.analysis.xsd XML Schema.

Please note some analysis types require certain object relations so please read the individual submission guides for more detail on what needs submitting for each analysis type and please register all required objects before beginning your analysis submission.

How to register/submit other datasets to ENA:

- Register a study
- Register samples
- Submit raw reads

### 8.2 Submission Options

All analyses require programmatic submission.

For an overview of how to submit programmatically, please see the documentation on Programmatic Submissions.

#### 8.2.1 Submitting Read Alignments

**Introduction**

To submit an analysis programmatically, two XML files must be generated to describe the submission.

- **Analysis XML** - used for describing the analysis you would like to submit
- **Submission XML** - tells ENA how to process this submission

These are then submitted to ENA through the secure HTTPS protocol using POST multipart/form-data according to RFC1867. Please see the general guide on Programmatic Submission for more information.

**Step 1: Create Analysis XML**

This XML is used for:

- Associating the analysis with other ENA objects
- Listing all files required for submission
- Describing metadata of the object

Here is an example of a read alignment analysis XML:

```xml
<?xml version="1.0" encoding="US-ASCII"?>
<ANALYSIS_SET>
  <ANALYSIS alias="AD0370_C_alignment">
    <TITLE>The Anopheles gambiae 1000 Genomes Project - Phase 1 - Alignment - Crosses</TITLE>
    <DESCRIPTION>Sequence alignments from the AR3 data release from the Anopheles gambiae 1000 genomes project. Alignments are in bam format and are presented for each of the 80 A. gambiae specimens comprising parents and progeny of four crosses.</DESCRIPTION>
  </ANALYSIS>
</ANALYSIS_SET>
```
In this example, the BAM file aligns reads from three sequencing runs to the reference genome ‘GCA_000005575’ and has three read group tags defined in its header that each represent one run:

<table>
<thead>
<tr>
<th>@RG</th>
<th>ID:8149_4_48</th>
<th>PL:ILLUMINA</th>
<th>PU:8149_4_48</th>
<th>LB:AD0370_C_</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-5557918</td>
<td>DS:AGPED1</td>
<td>SM:AD0370-C</td>
<td>CN:SC</td>
</tr>
<tr>
<td>@RG</td>
<td>ID:8177_1_48</td>
<td>PL:ILLUMINA</td>
<td>PU:8177_1_48</td>
<td>LB:AD0370_C_</td>
</tr>
<tr>
<td></td>
<td>-5557918</td>
<td>DS:AGPED1</td>
<td>SM:AD0370-C</td>
<td>CN:SC</td>
</tr>
<tr>
<td>@RG</td>
<td>ID:8177_2_48</td>
<td>PL:ILLUMINA</td>
<td>PU:8177_2_48</td>
<td>LB:AD0370_C_</td>
</tr>
<tr>
<td></td>
<td>-5557918</td>
<td>DS:AGPED1</td>
<td>SM:AD0370-C</td>
<td>CN:SC</td>
</tr>
</tbody>
</table>

The BAM header contains a single assembly and a reference sequence:

<table>
<thead>
<tr>
<th>@HD</th>
<th>VN:1.4</th>
<th>GO: none</th>
<th>SO: coordinate</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SN:2L</td>
<td>LN:49364325</td>
<td>UR:<a href="http://www.vectorbase.org/content/">http://www.vectorbase.org/content/</a></td>
</tr>
<tr>
<td></td>
<td>-anopheles-gamb</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>-pestchromosomesagamp3fagz</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>SP:Anopheles gambiae</td>
<td>AS:AgamP3</td>
<td>M5:a4da4bafa82830c0a418c5a42138377b</td>
</tr>
</tbody>
</table>

Defining the Analysis Type

The most distinguishing part of an analysis object is contained in the `<ANALYSIS_TYPE>` block. The content of this block determines the type of data the analysis should contain and how it will be validated by ENA after it has been submitted.

Analysis type `<REFERENCE_ALIGNMENT>` is used for submitting read alignments to ENA. For analyses of this type, you must reference any reference genomes or sequences used for the alignment.
Please note additionally that the reads used in the alignment should also already have been submitted following in-
structions in Submit Read Data.
Read alignment analyses must be associated with a study and can be associated with one or more samples.

**Associating with Other ENA Objects**

**Associating with the Study**

An analysis points to the study it is part of using the `<STUDY_REF>` element. This can be done either by using an accession:

```
<STUDY_REF accession="ERP123456"/>
```

or a name within the submitter’s account:

```
<STUDY_REF refname="mantis_religiosa"/>
```

**Associating with Samples**

An analysis can be associated with one or more samples using the `<SAMPLE_REF>` element either using an accession or alias to refer to the sample.

**Associating with Experiments and Runs**

An analysis can be associated with any number of experiments or runs using the `<EXPERIMENT_REF>` and `<RUN_REF>` elements. Again, either an accession or alias can be used in the reference.

**Preparing Files For Submission**

**Upload Data Files**

Please upload all data files required for submission.

Once the analysis has been submitted, all the data files described in the analysis XML will be moved from the Webin upload area into the archive.

You can upload your data files to the root directory of your upload area or you can create subdirectories and upload your files there.

**Describe Data Files for Submission**

You should then describe these data files in your analysis XML with the `<FILE>` element.

To describe files required for submission, the analysis object has a `<FILES>` block. This submits the data files into the archive.
For example:

```
<FILES>
  <FILE checksum="bafe0ed9be5c0f8515cdc4ac514d24af" checksum_method="MD5"
    filename="AD0370_C.bam" filetype="bam"/>
</FILES>
```

If the files are uploaded to the root directory then simply enter the file name in the Analysis XML when referring to it:

```
<FILE filename="a.bam" ... />
```

If the files are uploaded into a subdirectory (e.g. mantis_religiosa) then prefix the file name with the name of the subdirectory:

```
<FILE filename="mantis_religiosa/a.bam" ... />
```

**Adding Additional Metadata**

All other metadata used to describe the analysis can be provided using `ANALYSIS_ATTRIBUTE` elements in the XML:

```
<ANALYSIS_ATTRIBUTE>
  <TAG>library preparation date</TAG>
  <VALUE>2010-08</VALUE>
</ANALYSIS_ATTRIBUTE>
```

**Step 2: Create the Submission XML**

Once you have created your analysis XML, you need an accompanying submission XML in a separate file to tell ENA what actions you would like to take for your submission.

```
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <ADD/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```

The submission XML declares one or more Webin submission service actions. See the general guide on [Programmatic Submission](#) for more information.

In this case the action is `<ADD/>` which is used to submit new objects.

The XMLs can then be submitted programmatically, using CURL on command line or using the Webin submissions portal.

**Step 3: Submit the XMLs**

**Submit the XMLs Using CURL**

CURL is a Linux/Unix command line program which you can use to send the `analysis.xml` and `submission.xml` to the Webin submission service.
curl -u username:password -F "SUBMISSION=@submission.xml" -F "ANALYSIS=@analysis.xml" -d="https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit/"

Please provide your Webin submission account credentials using the *username* and *password*.

After running the command above a receipt XML is returned. It will look like the one below:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<RECEIPT receiptDate="2017-08-11T15:07:36.746+01:00" submissionFile="sub.xml" success="true">
  <ANALYSIS accession="ERZ0151578" alias="08-1736" status="PRIVATE"/>
  <SUBMISSION accession="ERA986371" alias="08-1736"/>
  <MESSAGES>
    <INFO>This submission is a TEST submission and will be discarded within 24 hours</INFO>
  </MESSAGES>
  <ACTIONS>ADD</ACTIONS>
  <ACTIONS>ADD</ACTIONS>
</RECEIPT>
```

**Submit the XMLs Using Webin submissions portal**

XMLs can also be submitted interactively using the Webin submissions portal. Please refer to the Webin submissions portal document for an example how to submit a study using XML. Other types of XMLs can be submitted using the same approach.

**The Receipt XML**

To know if the submission was successful look in the first line of the `<RECEIPT>` block. The attribute *success* will have value *true* or *false*. If the value is false then the submission did not succeed. In this case check the rest of the receipt for error messages and after making corrections, try the submission again.

If the success attribute is true then the submission was successful. The receipt will contain the accession numbers of the objects that you have submitted.

**Test and Production Services**

Note the message in the receipt:

```xml
<INFO>This submission is a TEST submission and will be discarded within 24 hours</INFO>
```

It is advisable to first test your submissions using the Webin test service where changes are not permanent and are erased every 24 hours.

Once you are happy with the result of the submission you can use the CURL command again but this time using the production service. Simply change the part in the URL from *wwwdev.ebi.ac.uk* to *www.ebi.ac.uk*:

```
curl -u username:password -F "SUBMISSION=@submission.xml" -F "ANALYSIS=@analysis.xml" -d="https://www.ebi.ac.uk/ena/submit/drop-box/submit/"
```

Similarly, if you are using the Webin submissions portal change the URL from *wwwdev.ebi.ac.uk* to *www.ebi.ac.uk*.
8.2.2 Submitting Sequence Annotations

Introduction

To submit an analysis programatically, two XML files must be generated to describe the submission.

- **Analysis XML** - used for describing the analysis you would like to submit
- **Submission XML** - tells ENA how to process this submission

These are then submitted to ENA through the secure HTTPS protocol using POST multipart/form-data according to RFC1867. Please see the general guide on Programmatic Submission for more information.

Step 1: Create Analysis XML

This XML is used for:

- Associating the analysis with other ENA objects
- Listing all files required for submission
- Describing metadata of the object

Here is an example of a sequence annotation analysis XML:

```xml
<ANALYSIS_SET>
  <ANALYSIS alias="YF3059">
    <TITLE>Y chromosome sequence STR analysis using lobSTR</TITLE>
    <DESCRIPTION>Y chromosome sequence STR analysis using lobSTR</DESCRIPTION>
    <STUDY_REF accession="ERP011288"/>
    <SAMPLE_REF accession="ERS1023190"/>
    <RUN_REF accession="ERR1198112"/>
    <ANALYSIS_TYPE>
      <SEQUENCE_ANNOTATION/>
    </ANALYSIS_TYPE>
    <FILES>
      <FILE filename="STR_for_YF03059_20151228.tab.gz" filetype="tab" checksum_method="MD5" checksum="9f2976d079c10b111669b32590d1eb3e"/>
    </FILES>
  </ANALYSIS>
</ANALYSIS_SET>
```

Defining the Analysis Type

The most distinguishing part of an analysis object is contained in the `<ANALYSIS_TYPE>` block. The content of this block determines the type of data the analysis should contain and how it will be validated by ENA after it has been submitted.

Analysis type `<SEQUENCE_ANNOTATION>` is for submitting sequence annotation files. These are usually tab files. Examples include gene count and OTU tables from metagenomic studies.

Sequence annotation analyses must be associated with a study and at most one sample.
Associating with Other ENA Objects

Associating with the Study

An analysis points to the study it is part of using the `<STUDY_REF>` element. This can be done either by using an accession:

```
<STUDY_REF accession="ERP123456"/>
```

or a name within the submitter’s account:

```
<STUDY_REF refname="mantis_religiosa"/>
```

Associating with Samples

An analysis can be associated with one or more samples using the `<SAMPLE_REF>` element either using an accession or alias to refer to the sample.

Associating with Experiments and Runs

An analysis can be associated with any number of experiments or runs using the `<EXPERIMENT_REF>` and `<RUN_REF>` elements. Again, either an accession or alias can be used in the reference.

Preparing Files For Submission

Upload Data Files

Please upload all data files required for submission.

Once the analysis has been submitted, all the data files described in the analysis XML will be moved from the Webin upload area into the archive.

You can upload your data files to the root directory of your upload area or you can create subdirectories and upload your files there.

Describe Data Files for Submission

You should then describe these data files in your analysis XML with the `<FILE>` element.

To describe files required for submission, the analysis object has a `<FILES>` block. This submits the data files into the archive.

For example:

```
<FILES>
  <FILE filename="STR_for_YF03059_20151228.tab.gz" filetype="tab" checksum_method="MD5" checksum="9f2976d079c10b111669b32590d1eb3e"/>
</FILES>
```

If the files are uploaded to the root directory then simply enter the file name in the Analysis XML when referring to it:
<FILE filename="a.tab.gz" />...

If the files are uploaded into a subdirectory (e.g. mantis_religiosa) then prefix the file name with the name of the subdirectory:

<FILE filename="mantis_religiosa/a.tab.gz" />

### Adding Additional Metadata

All other metadata used to describe the analysis can be provided using `ANALYSIS_ATTRIBUTE` elements in the XML:

```
<ANALYSIS_ATTRIBUTE>
  <TAG>library preparation date</TAG>
  <VALUE>2010-08</VALUE>
</ANALYSIS_ATTRIBUTE>
```

### Step 2: Create the Submission XML

Once you have created your analysis XML, you need an accompanying submission XML in a separate file to tell ENA what actions you would like to take for your submission.

```
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <ADD/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```

The submission XML declares one or more Webin submission service actions. See the general guide on Programmatic Submission for more information.

In this case the action is `<ADD/>` which is used to submit new objects.

The XMLs can then be submitted programmatically, using CURL on command line or using the Webin submissions portal.

### Step 3: Submit the XMLs

#### Submit the XMLs Using CURL

CURL is a Linux/Unix command line program which you can use to send the `analysis.xml` and `submission.xml` to the Webin submission service.

```
curl -u username:password -F "SUBMISSION=@submission.xml" -F "ANALYSIS=@analysis.xml" "https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit/"
```

Please provide your Webin submission account credentials using the `username` and `password`.

After running the command above a receipt XML is returned. It will look like the one below:
Submit the XMLs Using Webin submissions portal

XMLs can also be submitted interactively using the Webin submissions portal. Please refer to the Webin submissions portal document for an example how to submit a study using XML. Other types of XMLs can be submitted using the same approach.

The Receipt XML

To know if the submission was successful look in the first line of the <RECEIPT> block.

The attribute success will have value true or false. If the value is false then the submission did not succeed. In this case check the rest of the receipt for error messages and after making corrections, try the submission again.

If the success attribute is true then the submission was successful. The receipt will contain the accession numbers of the objects that you have submitted.

Test and Production Services

Note the message in the receipt:

<INFO>This submission is a TEST submission and will be discarded within 24 hours</INFO>

It is advisable to first test your submissions using the Webin test service where changes are not permanent and are erased every 24 hours.

Once you are happy with the result of the submission you can use the CURL command again but this time using the production service. Simply change the part in the URL from wwwdev.ebi.ac.uk to www.ebi.ac.uk:

```
curl -u username:password -F "SUBMISSION=@submission.xml" -F "ANALYSIS=@analysis.xml" -H "Content-Type: multipart/form-data" https://www.ebi.ac.uk/ena/submit/drop-box/submit/
```

Similarly, if you are using the Webin submissions portal change the URL from wwwdev.ebi.ac.uk to www.ebi.ac.uk.

8.2.3 Submitting PacBio Methylation Data
Introduction

To submit an analysis programmatically, two XML files must be generated to describe the submission.

- **Analysis XML** - used for describing the analysis you would like to submit
- **Submission XML** - tells ENA how to process this submission

These are then submitted to ENA through the secure HTTPS protocol using POST multipart/form-data according to RFC1867. Please see the general guide on Programmatic Submission for more information.

**Step 1: Create Analysis XML**

This XML is used for:

- Associating the analysis with other ENA objects
- Listing all files required for submission
- Describing metadata of the object

Here is an example of a PacBio methylation analysis XML:

```xml
<ANALYSIS_SET>
  <ANALYSIS alias="08-1736">
    <TITLE>Epigenomic analysis of Salmonella enterica 08-1736 from PacBio RS base-incorporation kinetic data</TITLE>
    <DESCRIPTION>Single-molecule read technologies allow for detection of epigenomic base modifications during routine sequencing by analysis of kinetic data during the reaction, including the duration between base incorporations at the elongation site (the "inter-pulse duration.") Methylome data associated with a closed de-novo bacterial genome of Salmonella enterica subsp. enterica serovar 4,5,12, i-str. 08-1736 was produced and submitted to the Gene Expression Omnibus.</DESCRIPTION>
    <STUDY_REF accession="SRP026480"/>
    <SAMPLE_REF accession="SRS454371"/>
    <ANALYSIS_TYPE>
      <SEQUENCE_ANNOTATION/>
    </ANALYSIS_TYPE>
    <FILES>
      <FILE filename="data-motifs.gff.gz" filetype="gff" checksum_method="MD5" checksum="7fd0cf4f550fd836758bfc242894a8fe"/>
      <FILE filename="data-motif_summary.csv.gz" filetype="tab" checksum_method="MD5" checksum="28e36d2792991de13aee0f377b774523"/>
      <FILE filename="data-modifications.csv.gz" filetype="tab" checksum_method="MD5" checksum="cebce127ade5bc04b0846b205151cbc9"/>
    </FILES>
  </ANALYSIS>
</ANALYSIS_SET>
```
Defining the Analysis Type

The most distinguishing part of an analysis object is contained in the `<ANALYSIS_TYPE>` block. The content of this block determines the type of data the analysis should contain and how it will be validated by ENA after it has been submitted.

Analysis type `<SEQUENCE_ANNOTATION>` is used for submitting PacBio methylation data to ENA.

Associating with Other ENA Objects

Associating with the Study

An analysis points to the study it is part of using the `<STUDY_REF>` element. This can be done either by using an accession:

```xml
<STUDY_REF accession="ERP123456"/>
```

or a name within the submitter’s account:

```xml
<STUDY_REF refname="mantis_religiosa"/>
```

Associating with Samples

An analysis can be associated with one or more samples using the `<SAMPLE_REF>` element either using an accession or alias to refer to the sample.

Associating with Experiments and Runs

An analysis can be associated with any number of experiments or runs using the `<EXPERIMENT_REF>` and `<RUN_REF>` elements. Again, either an accession or alias can be used in the reference.

Preparing Files For Submission

Upload Data Files

Please upload all data files required for submission.

Once the analysis has been submitted, all the data files described in the analysis XML will be moved from the Webin upload area into the archive.

You can upload your data files to the root directory of your upload area or you can create subdirectories and upload your files there.

Describe Data Files for Submission

You should then describe these data files in your analysis XML with the `<FILE>` element.

To describe files required for submission, the analysis object has a `<FILES>` block. This submits the data files into the archive.

For example:
If the files are uploaded to the root directory then simply enter the file name in the Analysis XML when referring to it:

```xml
<FILE filename="a.gff.gz" ...
```

If the files are uploaded into a subdirectory (e.g. mantis_religiosa) then prefix the file name with the name of the subdirectory:

```xml
<FILE filename="mantis_religiosa/a.gff.gz" ...
```

PacBio methylation data usually consists of a set of three files: modifications.csv, motif_summary.csv and motifs.gff. To learn more about what these files are and how to generate them, please refer to PacBio’s own documentation on the subject.

### Adding Additional Metadata

All other metadata used to describe the analysis can be provided using `<ANALYSIS_ATTRIBUTE>` elements in the XML:

```xml
<ANALYSIS_ATTRIBUTE>
    <TAG>library preparation date</TAG>
    <VALUE>2010-08</VALUE>
</ANALYSIS_ATTRIBUTE>
```

### Step 2: Create the Submission XML

Once you have created your analysis XML, you need an accompanying submission XML in a separate file to tell ENA what actions you would like to take for your submission.

```xml
<SUBMISSION>
    <ACTIONS>
        <ACTION>
            <ADD/>
        </ACTION>
    </ACTIONS>
</SUBMISSION>
```

The submission XML declares one or more Webin submission service actions. See the general guide on Programmatic Submission for more information.

In this case the action is `<ADD/>` which is used to submit new objects.

The XMLs can then be submitted programmatically, using CURL on command line or using the Webin submissions portal.

---

8.2. Submission Options 171
Step 3: Submit the XMLs

Submit the XMLs Using CURL

CURL is a Linux/Unix command line program which you can use to send the analysis.xml and submission.xml to the Webin submission service.

```bash
curl -u username:password -F "SUBMISSION=@submission.xml" -F "ANALYSIS=@analysis.xml" 
"https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit/"
```

Please provide your Webin submission account credentials using the username and password.

After running the command above a receipt XML is returned. It will look like the one below:

```xml
<?xml version="1.0" encoding="UTF-8"?>
<RECEIPT receiptDate="2017-08-11T15:07:36.746+01:00" submissionFile="sub.xml" success="true">
  <ANALYSIS accession="ERZ0151578" alias="08-1736" status="PRIVATE"/>
  <SUBMISSION accession="ERA986371" alias="08-1736"/>
  <MESSAGES>
    <INFO>This submission is a TEST submission and will be discarded within 24 hours</INFO>
  </MESSAGES>
  <ACTIONS>ADD</ACTIONS>
  <ACTIONS>ADD</ACTIONS>
</RECEIPT>
```

Submit the XMLs Using Webin submissions portal

XMLs can also be submitted interactively using the Webin submissions portal. Please refer to the Webin submissions portal document for an example how to submit a study using XML. Other types of XMLs can be submitted using the same approach.

The Receipt XML

To know if the submission was successful look in the first line of the <RECEIPT> block.

The attribute success will have value true or false. If the value is false then the submission did not succeed. In this case check the rest of the receipt for error messages and after making corrections, try the submission again.

If the success attribute is true then the submission was successful. The receipt will contain the accession numbers of the objects that you have submitted.

Test and Production Services

Note the message in the receipt:

```xml
<INFO>This submission is a TEST submission and will be discarded within 24 hours</INFO>
```

It is advisable to first test your submissions using the Webin test service where changes are not permanent and are erased every 24 hours.
Once you are happy with the result of the submission you can use the CURL command again but this time using the production service. Simply change the part in the URL from `wwwdev.ebi.ac.uk` to `www.ebi.ac.uk`:

```
curl -u username:password -F "SUBMISSION=@submission.xml" -F "ANALYSIS=@analysis.xml" -d "https://www.ebi.ac.uk/ena/submit/drop-box/submit/"
```

Similarly, if you are using the Webin submissions portal change the URL from `wwwdev.ebi.ac.uk` to `www.ebi.ac.uk`.

### 8.2.4 Submitting BioNano Maps

#### Introduction

To submit an analysis programmatically, two XML files must be generated to describe the submission.

- **Analysis XML** - used for describing the analysis you would like to submit
- **Submission XML** - tells ENA how to process this submission

These are then submitted to ENA through the secure HTTPS protocol using POST multipart/form-data according to RFC1867. Please see the general guide on Programmatic Submission for more information.

#### Step 1: Create Analysis XML

This XML is used for:

- Associating the analysis with other ENA objects
- Listing all files required for submission
- Describing metadata of the object

Here is an example of a BioNano map analysis XML:

```xml
<ANALYSIS_SET>
  <ANALYSIS alias="es_omd">
    <TITLE>Euclidium syriacum BioNano Optical Mapping data</TITLE>
    <DESCRIPTION>Euclidium syriacum Optical Mapping data produced by BioNano Genomics Irys System</DESCRIPTION>
    <STUDY_REF accession="ERP018601"/>
    <SAMPLE_REF accession="ERS1436420"/>
    <ANALYSIS_TYPE>
      <GENOME_MAP>
        <PROGRAM>IrysView</PROGRAM>
        <PLATFORM>BioNano</PLATFORM>
      </GENOME_MAP>
    </ANALYSIS_TYPE>
    <FILES>
      <FILE filename="Euclidium_syriacum.Run-01.bnx.gz" filetype="BioNano_native" checksum_method="MD5" checksum="ff9dd3a61d88092cb74ff8227ed725aa"/>
    </FILES>
  </ANALYSIS>
</ANALYSIS_SET>
```
Defining the Analysis Type

The most distinguishing part of an analysis object is contained in the <ANALYSIS_TYPE> block. The content of this block determines the type of data the analysis should contain and how it will be validated by ENA after it has been submitted.

Analysis type <GENOME_MAP> is for submitting BioNano optical map data to ENA. Optical maps must be associated with a study and a sample.

Associating with Other ENA Objects

Associating with the Study

An analysis points to the study it is part of using the <STUDY_REF> element. This can be done either by using an accession:

```xml
<STUDY_REF accession="ERP123456"/>
```

or a name within the submitter’s account:

```xml
<STUDY_REF refname="mantis_religiosa"/>
```

Associating with Samples

An analysis can be associated with one or more samples using the <SAMPLE_REF> element either using an accession or alias to refer to the sample.

Associating with Experiments and Runs

An analysis can be associated with any number of experiments or runs using the <EXPERIMENT_REF> and <RUN_REF> elements. Again, either an accession or alias can be used in the reference.

Preparing Files For Submission

Upload Data Files

Please upload all data files required for submission.

Once the analysis has been submitted, all the data files described in the analysis XML will be moved from the Webin upload area into the archive.

You can upload your data files to the root directory of your upload area or you can create subdirectories and upload your files there.

Describe Data Files for Submission

You should then describe these data files in your analysis XML with the <FILE> element.

To describe files required for submission, the analysis object has a <FILES> block. This submits the data files into the archive.
For example:

```xml
<FILES>
  <FILE filename="Euclidium_syriacum.Run-01.bnx.gz" filetype="BioNano_native"
    checksum_method="MD5" checksum="ff9dd3a61d88092cb74ff8227ed725aa"/>
</FILES>
```

If the files are uploaded to the root directory then simply enter the file name in the Analysis XML when referring to it:

```xml
<FILE filename="a.bnx" ... />
```

If the files are uploaded into a subdirectory (e.g. `mantis_religiosa`) then prefix the file name with the name of the subdirectory:

```xml
<FILE filename="mantis_religiosa/a.bnx" ... />
```

Note that the `filetype` must be "BioNano_native" and must refer to one of the following files:

- **CMAP**: The BioNano Genomics Irys .cmap file is a raw data view reporting a label site position within a genome map.
- **XMAP**: The BioNano Genomics Irys .xmap file is a cross-comparison derived from the alignment between an anchor .cmap file and a query .cmap file.
- **SMAP**: The BioNano Genomics Irys .smap file is a description of structural variations (SV) derived from the alignment between an anchor .cmap file and a query .cmap file.
- **BNX**: The BioNano Genomics Irys .bnx file is a raw data view of molecule and label information and quality scores per channel.
- **COORD**: The .coord file relates the coordinates of scaffolds in a hybrid assembly to a corresponding sequences.

For full details of the BioNano data files please refer to Bionano Solve documentation.

### Adding Additional Metadata

All other metadata used to describe the analysis can be provided using `ANALYSIS_ATTRIBUTE` elements in the XML:

```xml
<ANALYSIS_ATTRIBUTE>
  <TAG>library preparation date</TAG>
  <VALUE>2010-08</VALUE>
</ANALYSIS_ATTRIBUTE>
```

### Step 2: Create the Submission XML

Once you have created your analysis XML, you need an accompanying submission XML in a separate file to tell ENA what actions you would like to take for your submission.

```xml
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <ADD/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```
The submission XML declares one or more Webin submission service actions. See the general guide on Programmatic Submission for more information.

In this case the action is `<ADD/>` which is used to submit new objects.

The XMLs can then be submitted programmatically, using CURL on command line or using the Webin submissions portal.

**Step 3: Submit the XMLs**

**Submit the XMLs Using CURL**

CURL is a Linux/Unix command line program which you can use to send the `analysis.xml` and `submission.xml` to the Webin submission service.

```
curl -u username:password -F "SUBMISSION=@submission.xml" -F "ANALYSIS=@analysis.xml" \\ "https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit/"
```

Please provide your Webin submission account credentials using the `username` and `password`.

After running the command above a receipt XML is returned. It will look like the one below:

```
<?xml version="1.0" encoding="UTF-8"?>
<RECEIPT receiptDate="2017-08-11T15:07:36.746+01:00" submissionFile="sub.xml" success="true">
  <ANALYSIS accession="ERZ0151578" alias="08-1736" status="PRIVATE"/>
  <SUBMISSION accession="ERA986371" alias="08-1736"/>
  <MESSAGES>
    <INFO>This submission is a TEST submission and will be discarded within 24 hours</INFO>
  </MESSAGES>
  <ACTIONS>ADD</ACTIONS>
  <ACTIONS>ADD</ACTIONS>
</RECEIPT>
```

**Submit the XMLs Using Webin submissions portal**

XMLs can also be submitted interactively using the Webin submissions portal. Please refer to the Webin submissions portal document for an example how to submit a study using XML. Other types of XMLs can be submitted using the same approach.

**The Receipt XML**

To know if the submission was successful look in the first line of the `<RECEIPT>` block.

The attribute `success` will have value `true` or `false`. If the value is false then the submission did not succeed. In this case check the rest of the receipt for error messages and after making corrections, try the submission again.

If the success attribute is true then the submission was successful. The receipt will contain the accession numbers of the objects that you have submitted.

**Test and Production Services**

Note the message in the receipt:
It is advisable to first test your submissions using the Webin test service where changes are not permanent and are erased every 24 hours.

Once you are happy with the result of the submission you can use the CURL command again but this time using the production service. Simply change the part in the URL from wwwdev.ebi.ac.uk to www.ebi.ac.uk:

```
curl -u username:password -F "SUBMISSION=@submission.xml" -F "ANALYSIS=@analysis.xml" -L "https://www.ebi.ac.uk/ena/submit/drop-box/submit/
```

Similarly, if you are using the Webin submissions portal change the URL from wwwdev.ebi.ac.uk to www.ebi.ac.uk.

### 8.2.5 Submitting Taxonomic Reference Sets

**Introduction**

The Taxonomic Reference Set analysis class is intended as a stable means of systematically capturing datasets from established reference databases in a standard format. It is then possible to convey them to dependent resources for use in taxonomic assignment of metabarcoding sequences.

The format has three components:

- **Metadata Record**: Some general information about the dataset as a whole
- **Sequence Record**: A multi-FASTA containing the sequence to be archived, along with identifiers
- **Sequence Metadata Table**: A TSV file containing information about the origins of the sequences, matched by identifiers

The three components together comprise a Taxonomic Reference Set. The format of each item is described in detail below.

Before you can submit a Taxonomic Reference Set, you must register a study.

This analysis class can only be submitted through our Webin-CLI tool.

This page will describe each of the three components of a taxonomic reference set, and then provide instruction on how to submit this analysis type.

**Submission Content**

**Metadata Record**

All Webin-CLI submissions require a metadata record in the form of a manifest file. This describes the dataset as a whole, not the individual sequences.

The following fields are used for Taxonomic Reference Set Submission:

- **Name** (*MANDATORY*) - A short descriptive name for your dataset
- **Study** (*MANDATORY*) - A valid INSDC Study accession
- **Description** (*MANDATORY*) - A description of the dataset
- **Taxonomy System** (*MANDATORY*) - The database or authority defining the taxon names used in your metadata table. ‘NCBI’ or ‘NCBI Taxonomy’ is preferred
• **Taxonomy System Version** *(OPTIONAL)* - The version of the taxonomy system defined above, if applicable

• **FASTA** *(MANDATORY)* - The gzipped FASTA file containing your sequence records

• **TAB** *(MANDATORY)* - The gzipped TSV file containing your sequence metadata

• **CUSTOMFIELD** *(OPTIONAL)* - Custom headers you may have included in your sequence metadata table

It should be clear from the **name** and **description** what the theme of your dataset is (e.g. ITS1 sequences, 16S sequences, a specific clade). The name must always be unique among submissions from your account.

The **taxonomy system** field should specify which authority your taxon names and lineages conform to. Values ‘NCBI’ or ‘NCBI Taxonomy’ are preferred and are mandatory if you intend to include NCBI taxon IDs. However, any database may be specified here. You can clarify a version/release number with the **taxonomy system version** if you wish.

You may include **custom field** rows in the metadata record. These define and describe additional headers used in your sequence metadata table. Additional table headers and custom field entries must exactly match each other. Separate the header and description with a colon (e.g. Name1:Description1).

An example manifest follows:

```plaintext
NAME 16S Database Sequences
STUDY ERP999999
DESCRIPTION Set of reference 16S sequences as collected by 16S Database
TAXONOMY_SYSTEM NCBI
FASTA 16s_db_seqs.fasta.gz
TAB 16s_db_info.tsv.gz
CUSTOMFIELD header_name:description of column purpose
```

Download a blank manifest for your use here: tax_ref_set_example_manifest.txt

**Sequence Record**

The sequence data component of the submission, this is a multi-FASTA file containing marker sequences, such as ITS1 or 16S. It should have been made clear in the metadata record what to expect here.

The ID lines must begin with a sequence identifier unique to each record and matching an identifier in the sequence metadata table. It is acceptable to use an INSDC accession here. If you wish to include further details about the sequence here, separate them from the identifier with a ‘|’ character:

```
>ITS1DB00588027|162468|Steinernema abbasi|ITS1 located by ENA annotation, 267bp
```

Any valid nucleotide character is accepted, blank lines will not be accepted. Sequence lines must conform to the regular expression:

```
^[ATCGactgRYSKMBDHVryswkmbdhvNn]*$
```

The ID and sequence must each occupy a single line. An odd number of lines in a FASTA file is considered an error.

**Sequence Metadata Table**

This is a TSV-format file containing metadata for each individual sequence. Six mandatory columns are defined. All six must always be present, but some may contain null values:

• **local_identifier** *(NOT NULL)* - Must exactly match an identifier in the FASTA

• **insdc_sequence_accession** *(null allowed)* - Must be a valid INSDC sequence accession
• **insdc_sequence_range** (*null if INSDC accession is*) - Must be a valid INSDC location string (e.g. “<21..<235”, regex: “^<?d+.>?$”)

• **local_organism_name** (*NOT NULL*) - Binomial name of the species the sequence identifies

• **local_lineage** (*null allowed*) - Full lineage of the species named in the local_organism_name column

• **ncbi_tax_id** (*conditional*) NCBI Tax Id is mandatory if using NCBI Taxonomy, disallowed if not

The **INSDC sequence range** is optional, but must be null if the **INSDC sequence accession** is null.

If NCBI taxonomy is specified as the **taxonomic system**, you must provide **NCBI tax ID** for every entry. Otherwise, this column must always be null. If Tax IDs are included, they must match the organism name.

In addition to the mandatory headers, you may have chosen to define additional headers in the metadata record. If so, they must be included in this table.

Download a blank table for your use here: tax_ref_set_example_table.tsv

### Submitting The Files

The files must be submitted using Webin-CLI. For an introduction to this tool please [click here](#).

Webin-CLI includes **-test** and **-validate** options. **-validate** may be used to have the Webin-CLI validator run check your files without submitting. **-test** may be used to make your submission to our test server, which is wiped clean every 24 hours. Note that a newly submitted study cannot be referenced on the test server until 24 hours after its submission. You are advised to use these prior to submission to help identify problems with the files.

Webin-CLI requires the following options from you:

- Your Webin username, resembling ‘Webin-xxxxx’: **-username**
- Your Webin password: **-password**
- The type of submission, ‘taxrefset’ in this case: **-context**
- Name and path to your metadata record file: **-manifest**
- Instruction, either **-validate** or **-submit**
- Optionally, you may use **-test** to submit to the test server

An example command may resemble:

```
webin-cli -username Webin-99999 -password xxxxxx -context taxrefset -manifest trs_submission.txt -validate
```

When ready, switch **-validate** for **-submit**

### 8.3 Analysis File Groups

For each analysis type, only specified pre-defined groups of files are allowed in a submission. The allowed file group(s) for each analysis type are enumerated in the below table. There may be multiple allowed groups for each analysis type, and the amount of files may be a range rather than a single integer.

**Using This Table:** Having identified the analysis type you wish to submit, find it in the table and consider all file groups available for it. A single file group will occupy multiple lines if it includes multiple types of file. Not all file types will be mandatory for a file group, which will be indicated by the ‘Allowed Quantity’ column showing that 0 is an acceptable quantity of files for a given type.
<table>
<thead>
<tr>
<th>Analysis Type</th>
<th>File Group Name</th>
<th>File Format</th>
<th>Allowed Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Genome Map</td>
<td>Bionano Native</td>
<td>BIONANO_NATIVE</td>
<td>1+</td>
</tr>
<tr>
<td>Reference Alignment</td>
<td>Single BAM</td>
<td>BAM</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Single CRAM</td>
<td>CRAM</td>
<td>1</td>
</tr>
<tr>
<td>PacBio Methylation</td>
<td>Sequence Methylation</td>
<td>GFF</td>
<td>1</td>
</tr>
<tr>
<td>Sequence Annotation</td>
<td>Sequence Methylation</td>
<td>TAB</td>
<td>0-2</td>
</tr>
<tr>
<td></td>
<td>Single TAB</td>
<td>TAB</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Single FASTA</td>
<td>FASTA</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Single BED</td>
<td>BED</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Single GFF</td>
<td>GFF</td>
<td>1</td>
</tr>
<tr>
<td>Taxonomic Reference Set</td>
<td>Taxonomic Reference Set</td>
<td>FASTA</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Taxonomic Reference Set</td>
<td>TAB</td>
<td>1</td>
</tr>
</tbody>
</table>
Welcome to the general guide for the European Nucleotide Archive data discovery and retrieval. Please take some time to view this introduction and explore all the options available for our data retrieval services.

9.1 Viewing and Exploring ENA Records

The table below summarises the domains of data held within ENA and example records that are archived within each domain and displayed within the ENA Browser. Please see our How To guide on exploring an ENA project for an example of how to navigate through an ENA Project in the browser:

9.1.1 How to Explore an ENA Project

A Project registered with the European Nucleotide Archive represents a real-life research project and is a way to share the data generated as part of that research openly and publicly with the rest of the research community.

In this guide we are going to use the example project PRJEB1787 and explore the motivations, contents and related records with this project. Different projects will contain different types of records: raw reads, assemblies or different analyses specific to that research. This guide is not comprehensive of all types of ENA research project but should give a basis of how to perform the exploration yourself in order to get the most out of the ENA research project.

Viewing the Project in The Browser

Firstly - navigate to the ENA project. This can be done by typing the accession PRJEB1787 into the box where it says ‘Enter accession’ on the top right of the ENA browser homepage - this will take you directly to the ENA record. Alternatively, you can add the accession to the ENA Browser url below:

https://www.ebi.ac.uk/ena/browser/view/<accession>

e.g.:

https://www.ebi.ac.uk/ena/browser/view/PRJEB1787
Once the page has fully loaded, the project should look like this:

The Navigation Box

The main way to navigate through an ENA project is using the navigation box in the top right corner. This will allow you to show/hide different types of information about that project. The contents of the navigation box will vary depending on the contents of the project. For example, some projects will have Read Files, others may have Analysis Files. Some projects have a large number of different components, so be patient if it takes a while to load.

This is the navigation box for project PRJEB1787:

This navigation box allows you to:

Navigate directly through the components of the project in the ‘Navigation’ tab. This also allows you to explore any cross-references related to this project (links to this record or a related record in other resources).

There is also the option to explore the uploaded ‘Read Files’ generated as part of that research project.
You can also explore any ‘Additional attributes’ that describe this project, links to any associated ‘Publications’ as well as a summary of all ‘Related ENA Records’ with the project.

**Umbrella Projects**

You may also come across umbrella projects. Navigate now to the [Tara Oceans](#) umbrella project [PRJEB402](#).

In some cases, such as this one, you will see an umbrella icon in the navigation box. This indicates that this project is an **umbrella project**. Instead of holding data, this project provides an umbrella to group together multiple projects that are part of the same research motivation or collaboration. You can see a list of all the projects within an umbrella project when viewing the ‘Component Projects’ tab.

Click on the ‘Component Projects’ tab and have a look at the contents here. You should see project [PRJEB1787](#) listed as one of the components. Don’t navigate back to project [PRJEB1787](#) yet. We’ll look first at the research motivations of the Tara Oceans project.

**Exploring Associated Publications**

A good place to start when looking into any ENA project, is to see if the project has any associated publications. Associated publications can help you understand the research motivations and conclusions that led to the data generation within that project. For example, as part of the Tara Oceans umbrella project we are looking at, you can read more about the Tara-Oceans scientific expedition (2009-2012) in the ‘Publications’ tab.

Publications are available in a table display with links to the article’s DOI or Europe PMC.

There are links to ‘Publications in which the generation of this record is described’ which can be useful when looking into the methods used for this piece of research. There are also links to ‘Publications citing this record’ which can give insight into the influence these data have had on further research.
Once you have finished exploring the associated literature, navigate back to PRJEB1787 so we can explore the data that was generated as part of this particular Tara Oceans component project further.

What Does the Project Contain?

The next questions to ask yourself is - what are the contents of this ENA Project?

To explore the full contents of a project, you can see overviews of all records within the ‘Navigation’ and ‘Related ENA Records’ tabs. For example, click on the ‘Navigation’ tab:

Here we can see a list of all the records that are part of the ENA Project. You can click the Help button on the far right of the tab for more information on the organisation of an ENA project and what is included here. In this example, you can see that the project holds a number of Samples, Experiments and Runs (if you are not familiar with the different types of ENA records, have a look at the table in the general guide). This means this project holds raw read data files. We can also see ‘Europe PMC’ records, ‘MGnify’ and ‘MarCat’ links. These are cross-references - records which are linked to the ENA record in other resources. Let’s click on the MGnify (EMBL-EBI metagenomics service) cross

184 Chapter 9. General Guide on ENA Data Retrieval
reference **MGYS00000410** as an example. You’ll see that this takes you to a MGNify study which allows you to explore the analyses performed by MGNify which were done using the reads in project **PRJEB1787**.

If you would like to learn more about exploring cross-references in ENA, please see our guide on exploring cross-references for further information.

Feel free to explore the other cross-references available for this project or look into the MGNify analyses then navigate back to the Navigation tab in **PRJEB1787**.

We know that project **PRJEB1787** contains Samples, Experiments and Runs, which we know means that this project contains Read Files. So have a look into the Read Files tab to see a table of all the data generated as part of this project.

Have a look at what additional metadata you can view in the table using the ‘Show selected columns’ drop down. Also, have a look at what files you can download as part of this project.

**Looking into Sample Records**

If you want to know any specific details about the samples that were collected (and sequenced) during the research project, you can look into the Sample records that are within the Project.

Sample records contain the most extensive metadata. These explain how the data was collected and can be used to understand and better interpret the contents of any related data files. For example, navigate to sample **SAMEA2591108** (ERS478017) either in the Read Files table or through the Navigation tab.

From within this sample record, you can still see the read files that were part of the original project. But from here you can also see other uses of the data associated with this sample. In particular, you’ll notice that you can now see an Analysis Files tab:

When you click on the Analysis Files tab, you can see a number of analyses of the raw read data. Within this table you can also see that these are linked to different ENA Projects:
By looking into the sample record you are able to further explore re-uses of that sample or the raw data associated with it.

**“Additional Attributes”**

The ‘Additional Attributes’ tab is one of the most important tabs. This tab holds all the metadata associated with that record. It is invaluable for details on sample records.

Navigate to the Additional Attributes tab:

Here you can see all the details of the sample collection event - when it was collected, where it was collected, at what depth etc. This information can be incredibly useful to provide context to the associated data files. It is also the metadata used to help search across records.
Sample Checklists

If you look at the main descriptions of the sample at the top of the page you will see there is a Checklist record listed (ERC000030). A sample checklist is a list of mandatory, recommended or optional metadata that was required during registration of that sample. Checklists help standardise the metadata of samples to make it easier to interpret and search across them.

These checklists are developed with the help of research communities themselves to encourage submitters to provide at least the minimum information for their samples to be findable and re-usable for further research. You can see the full list of checklists here.

If you want to search for a particular type of sample (e.g. a soil sample), you can search across the archive by the checklist (e.g. searching for samples that used the GSC MIxS soil checklist). See our guide on performing an advanced search for details on how to do this.

Keep an Eye Out For Help

The best way to get the most of an ENA project is to explore - see what data and metadata are available and delve into any other research linked to these.

The ENA browser has several pop up Help icons which can aid you in navigating the site. Keep an eye out for any help that can assist you in exploring or interpreting records in ENA.
<table>
<thead>
<tr>
<th>Data Domain</th>
<th>Description</th>
<th>Example Records</th>
</tr>
</thead>
<tbody>
<tr>
<td>Projects/Studies</td>
<td>Contains information on a biological research project. This holds all the data generated as part of this research</td>
<td>PRJEB1787 (ERP001736)</td>
</tr>
<tr>
<td>Samples</td>
<td>Represents biological samples collected and sequenced in real life</td>
<td>SAMEA2620084 (ERS488919)</td>
</tr>
<tr>
<td>Reads (Runs/Experiments)</td>
<td>Hold raw read files and sequencing methods</td>
<td>ERR1701760 ERX1772048</td>
</tr>
<tr>
<td>Analyses</td>
<td>Hold results files of analyses performed on sequencing data and analysis methods</td>
<td>ERZ1195979</td>
</tr>
<tr>
<td>Contig set</td>
<td>Hold contig sets generated as part of a genome or transcriptome assembly.</td>
<td>CABHOY010000000.1</td>
</tr>
<tr>
<td>Assemblies</td>
<td>Represents an entire genome assembly and holds any contig sets or sequence records generated as part of the assembly</td>
<td>GCA_000001405.28</td>
</tr>
<tr>
<td>Assembled/Annotated Sequences (*)</td>
<td>Any sequence records from coding or non-coding regions to full assembled chromosomes</td>
<td>CM000667.2</td>
</tr>
<tr>
<td>Taxon</td>
<td>The sequenced organism or metagenome of a sample</td>
<td>Taxon:9606</td>
</tr>
<tr>
<td>Sample Checklist</td>
<td>The checklist of metadata that the sample was registered with</td>
<td>ERC0000013</td>
</tr>
</tbody>
</table>
* Assembled and annotated sequence records fall into different data classes. Read more about the different classes of sequences here.

## 9.2 Search and Retrieval

You can search across the ENA browser in a number of ways:

- free text search
- sequence similarity search
- cross-reference search
- advanced search

The advanced search in the browser provides a simple interface for building more complex search queries that can be saved and run again with Rulespace. See our step by step guide on how to use the advanced search for examples on how to build queries and how to use Rulespace:

### 9.2.1 How To Perform An Advanced Search

The ENA Advanced Search interface is a powerful way to search the archive for datasets which meet criteria of interest to you as a user. The below guide will walk you through the interface using examples. Use this to familiarise yourself with the interface before conducting your own searches.

### Example Uses

Below are two example cases you can follow to better understand the advanced search interface:

- **Find Neanderthal Read Data**
- **Find Samples Based On Checklists**

### Find Neanderthal Read Data

Locate all read data submitted for Neanderthals

1. Navigate to the advanced search browser: [https://www.ebi.ac.uk/ena/browser/advanced-search](https://www.ebi.ac.uk/ena/browser/advanced-search)
2. The interface first asks you to select the datatype you want to search for. Use the ‘?’ icon next to the drop-down box to learn more about the options. We are searching for read data, so select ‘Raw Reads’.

3. Click ‘Next’

4. You now have opportunity to apply filters which will limit the set of reads which are provided

5. Use the blue buttons on the left to select different filter categories, or use the ‘Type to filter query params’ box to search for a specific parameter

6. In the above box, we have selected the ‘Taxonomy and related’ category, and chosen ‘NCBI Taxonomy’ from the dropdown

7. All submissions to ENA must use a valid species-level taxon from the NCBI taxonomy database. Begin typing and suggestions appear. We have put ‘Neanderthal’. Notice multiple suggestions are given; in this case, they are synonyms, as indicated by their matching taxon IDs (63221)
8. We have entered all the parameters we want, so click ‘Next’ twice to get to the Fields selection box.

9. Now click the ‘Manually select fields’ box.

10. For reads, there are three pre-defined field sets you can use: FASTQ Files gives you the archive-generated standardised FASTQ format, SRA Files gives file versions suitable for use with NCBI’s SRA Toolkit, while Submitted Files gives you copies of the files originally submitted by the user.

11. You can use the ‘Available Fields’ box to choose any fields you wish to see included in the table of results. Not all of these will always be relevant. For this search, we have selected the FASTQ Files preset, but if you skip this step you will still be shown a list of accessions with descriptions, by default.

12. When you are satisfied with the fields you have chosen, click ‘Search’ to see the results of your query.
Find Samples Based On Checklists

ENA sample objects represented physical samples used in sequencing experiments. Users must conform to a checklist of values when submitting samples. We can leverage this in searches to find samples belonging to a checklist which describes materials of interest to us.

Follow the below guide to find samples registered using the checklist named ‘GSC MIxS human gut’, which describes human gut microbiome samples

1. Navigate to the advanced search browser: https://www.ebi.ac.uk/ena/browser/advanced-search

2. The interface first asks you to select the datatype you want to search for. Use the ‘?’ icon next to the drop-down box to learn more about the options. Select the option simply named ‘Samples’

3. Click ‘Next’

4. Categories of search parameters are found by clicking the blue boxes to the left of the page. Select the ‘Database record’ category now

5. In the beige box, use the dropdown box to select the ‘Checklist’ option

6. A new dropdown box will appear: use this to select the checklist you are interested in - ‘GSC MIxS human gut’ in this case

7. Notice the accession number of the checklist, ‘ERC000015’, is added to the ‘Query’ box
8. Now click ‘Next’ twice to get to the Fields selection box
9. You can use the ‘Available Fields’ box to choose any fields you wish to see included in the table of results. Not all of these will always be relevant
10. When you are satisfied with the fields you have chosen, click ‘Search’ to see the results of your query

**Rulespace**

Rulespace is a tool which allows you save your searches, give them names, rerun them and share them with other people. Learn more about it in the Rulespace Guide.

The ENA Browser also provides different means to download data from the archive whether its XML ENA records, a tabulated summary of metadata resulting from a search or sequencing data files submitted as part of a research project. See our guide on file download for details on how to use our data retrieval services to download data from the archive:

### 9.2.2 How to Download Data Files

Providing users with the ability to download submitted data for further analysis purposes is a key part of ENA’s mission. Files are therefore made available through a public FTP server. Here you can learn how this server is structured, and how to download read and analysis files.

**FTP Structure**

The root address of the FTP server containing all read and analysis data is:

```
ftp://ftp.sra.ebi.ac.uk/vol1/
```

Meanwhile, assembled and annotated sequence data can be found at:

```
ftp://ftp.ebi.ac.uk/pub/databases/ena/
```

Any file you download from ENA will come from one of these two FTP servers. Their content and structures are described in detail at the below pages:

**SRA FTP Structure**

The root address of the SRA FTP server is:

```
ftp://ftp.sra.ebi.ac.uk/vol1/
```

This server contains all raw read data submitted to ENA and INSDC, as well a subset of analysis data.

This server explicitly excludes genome assemblies of all kinds, metagenome assemblies, transcriptome sequence assemblies, template sequences, and other short pieces of annotated sequence. This type of data can be found in ENA FTP. Analyses found in SRA FTP include variant data in VCF format, methylation data, and taxonomic reference sets.

Please see below for information on the structure of the FTP as it pertains to the two types of data stored in SRA FTP:

- **Reads FTP Structure**
- **Analysis FTP Structure**
Reads FTP Structure

For most reads presented by ENA, there are three kinds of file available:

- **Submitted files** are identical to those submitted by the user
- **FASTQ files** are archive-generated files generated according to a standardised format (learn more about this format)
- **SRA files** are in a format designed to work with NCBI’s SRA Toolkit

Each of the three file types has its own directory on the FTP server. A folder exists for every run, which is named with the accession of the run, e.g. files for run ERR164407 are in a directory named ERR164407. These run accession directories are organised into parent directories named with their first 6 characters. For ERR164407 this is ‘ERR164’. This structure is repeated across all three file types

For example, the run ERR164407 would be found in a directory resembling:

```
ftp://ftp.sra.ebi.ac.uk/vol1/<file-type>/ERR164/ERR164407
```

Specifics of how to access each read type are available below.

Submitted Read Files

The submitted read file is always present for all runs submitted to ENA. If a run was originally submitted to another INSDC database (NCBI SRA, DDBJ) then a file of this category will not be available.

ENA submitted files are available in the ‘run’ directory, e.g.:

```
ftp://ftp.sra.ebi.ac.uk/vol1/run/<accession-prefix>/<full-accession>/
```

Using ERR164407 as an example:

```
ftp://ftp.sra.ebi.ac.uk/vol1/run/ERR164/ERR164407/
```

Navigating to this directory in your browser will reveal a file named ‘APY_COTS_GPUCM0403.sff’.

Archive-Generated Read Files

In most cases ENA generates one or more FASTQ files for each run, which follow a specific format. For information on this format, and when it is not available, please see our page on Archive Generated FASTQ Files.

ENA archive-generated files are available in the ‘fastq’ directory, e.g.:

```
ftp://ftp.sra.ebi.ac.uk/vol1/fastq/<accession-prefix>/<full-accession>/
```

Using ERR164407 as an example:

```
ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR164/ERR164407/
```

Navigating to this directory in your browser will reveal a file named ‘ERR164407.fastq.gz’. An archive-generated FASTQ file always has a name which contains the run accession.
SRA Read Files

SRA format files are provided for use with NCBI's SRA Toolkit. SRA files are made available in the 'err' directory, e.g.:

ftp://ftp.sra.ebi.ac.uk/vol1/err/<accession-prefix>/<full-accession>/

Using ERR164407 as an example:

ftp://ftp.sra.ebi.ac.uk/vol1/err/ERR164/ERR164407/

Analysis FTP Structure

The ENA analysis object can hold a variety of different data types. It is designed to be flexible and extensible to allow quick implementation of new data types.

Analyses come in two types:

- Processed analyses include materials such as genome assemblies, transcriptome assemblies, MAGs, and other annotated sequences: these are stored in the ENA FTP.
- Simple analyses include various types described in our page on How To Submit Other Analyses as well as primary and binned metagenomes are stored in the SRA FTP.

These simple analyses undergo some validation but are mostly presented in the same format the user submitted them: this is the type of analysis described in this section.

The variation in types of analyses prevents the sort of processing and organisation given to reads, as described above. Analyses are stored in directories named with the analysis accession, e.g. ERZ1195979. These directories are stored within parent directories named with the first six characters of their accession. Therefore, files for analysis ERZ1195979 can be found at:

ftp://ftp.sra.ebi.ac.uk/vol1/ERZ119/ERZ1195979/

This area of the FTP server includes primary and binned metagenomes, as well as analysis classes described in our page on How To Submit Other Analyses.

ENA FTP Structure

The root of the ENA FTP server is at:

ftp://ftp.ebi.ac.uk/pub/databases/ena/

This server contains short pieces of assembled/annotated sequence dating back to the early 80s, as well as larger scale data types including genome assemblies, MAG/SAG assemblies, and transcriptome sequence assemblies. This server does not contain other data types such as raw reads, which can be found in the SRA FTP server.

It is expected that users will find relevant data through the various search interfaces available from ENA, rather than directly interacting with and navigating the FTP server. This page therefore does not provide an exhaustive explanation of the server’s structure, but serves to briefly describe what you will find in some of its most important subdirectories.

For some, but not all categories, suppressed data is kept separate from public.

Most data is available in EMBL flat file format, with some also available as FASTA.
Doc
Information on the INSDC feature table format, with versions dating back to 1995
ftp://ftp.ebi.ac.uk/pub/databases/ena/doc/

Genome Collections
An XML file containing cumulative information on all genome collection accessions
ftp://ftp.ebi.ac.uk/pub/databases/ena/genome_collections/

Assembly
Plaintext reports on the content of genome assemblies, organised by genome collection accession (GCA)
ftp://ftp.ebi.ac.uk/pub/databases/ena/assembly/

WGS
Whole genome shotgun contig-level assembly, one multi-record flatfile and one multi-FASTA file per sequencing project
ftp://ftp.ebi.ac.uk/pub/databases/ena/wgs/

TSA
Transcriptome shotgun assembly records, one multi-record flatfile and one multi-FASTA file per sequencing project
ftp://ftp.ebi.ac.uk/pub/databases/ena/tsa/

Sequence
Generalised sequence category, organised by sequence type and taxon class
ftp://ftp.ebi.ac.uk/pub/databases/ena/sequence/

Coding
Protein-coding sequences, organised by sequence type and accession number
ftp://ftp.ebi.ac.uk/pub/databases/ena/coding/

Non-Coding
Non-protein-coding sequences, organised by sequence type and accession number
ftp://ftp.ebi.ac.uk/pub/databases/ena/non-coding/

rRNA
Ribosomal RNA sequences, organised by sequence type and accession number
ftp://ftp.ebi.ac.uk/pub/databases/ena/rRNA/

TLS
Targeted locus study sequences, one multi-record flatfile and one multi-FASTA file per sequencing project
ftp://ftp.ebi.ac.uk/pub/databases/ena/tls/
Downloading Files

ENA provides numerous ways to access the data it hosts, suiting a range of use-cases and computational ability levels. These are described below, ranked from low to high, based on how much computational ability might be required:

- Using ENA Browser
- Using ENA FTP Downloader
- Using Globus
- Using enaBrowserTools
- Using wget
- Using FTP Client
- Using Aspera

**Note:** Most directories contain a ‘.md5’ file. You can calculate the MD5 value for a file you have downloaded and compare it with the relevant .md5 file to confirm it has been transferred in full.

**Using ENA Browser**

The ENA Browser is our website, from which you can get information about ENA, as well as accessing all the data we have public. Visit us here:

https://www.ebi.ac.uk/ena/browser/home

You can go to any accession by entering it into the ‘Enter accession’ box at the link above. If, for example, you see an ENA accession referenced in a paper, you can see the data for yourself in this way. Once there, you can download any associated files by clicking the relevant links. For more information on how to explore a record in ENA, please visit our guide on How to Explore an ENA Project

**Using ENA FTP Downloader**

The ENA FTP File Downloader is an application you can download from GitHub. Given an accession, this program will present a list of associated files you can download. Alternatively, you can provide a query from our Advanced Search API or Portal API to perform a bulk download of all files for a given set of criteria. Learn more about these APIs from our guide on How to Access ENA Programmatically.

**Using Globus**

Globus provides a more user-friendly, feature-rich directory interface for interacting with the FTP server. Files can be downloaded through Globus ‘Shared EMBL-EBI public endpoint’ endpoint from the ‘/gridftp/ena’ subfolder:
Using *enaBrowserTools*

enaBrowserTools is a set of Python-based utilities which can be found here. These are simple-to-run scripts which allow accession-based data download commands with the option to create more complex commands. Read more about this page in the *enaBrowserTools Guide*.

Using *wget*

`wget` is a simple command line tool, ubiquitously available in Linux and Mac releases. A file can be downloaded with `wget` simply by specifying its location:

```
$ wget ftp://ftp.sra.ebi.ac.uk/vol1/fastq/ERR164/ERR164407/ERR164407.fastq.gz
```

Using FTP Client

Command-line FTP clients allow you to interactively explore the FTP server and download data to your local computer. When asked for a username, use ‘anonymous’. When asked for a password, press the enter key to skip this.
In the above example, the `cd` command is used to ‘change directory’ to the required directory. Then, the ‘get’ command is used to specify the file of interest. At any time, you can use ‘ls’ to view the content of the current directory. The command ‘pwd’ can be used to identify what the current directory is.

**Using Aspera**

Aspera ascp command line client can be downloaded from Aspera. Please select the correct version for your operating system. The ascp command line client is distributed as part of the Aspera connect high-performance transfer browser plug-in.

Following are some examples of how Aspera may be used to download ENA data:

**Unix**

```
ascp -QT -l 300m -P33001 -i path/to/aspera/installation/etc/asperaweb_id_dsa.openssh \
era-fasp@fasp.sra.ebi.ac.uk:vol1/fastq/ERR164/ERR164407/ERR164407.fastq.gz \
local/target/directory
```

**Mac OSX**

```
ascp -QT -l 300m -P33001 -i path/to/aspera/installation/asperaweb_id_dsa.openssh \
era-fasp@fasp.sra.ebi.ac.uk:vol1/fastq/ERR164/ERR164407/ERR164407.fastq.gz \ 
local/target/directory
```

**Windows**

```
"%userprofile%\AppData\Local\Programs\Aspera\Aspera Connect\bin\ascp" ^
-QT -l 300m -P33001 -i ^
"%userprofile%\AppData\Local\Programs\Aspera\Aspera Connect\etc\asperaweb_id_dsa. ^
    openssh" ^
era-fasp@fasp.sra.ebi.ac.uk:vol1/fastq/ERR164/ERR164407/ERR164407.fastq.gz ^
local\target\directory
```

**Downloading Private Files**

E.g. If you want to use aspera to download a non-public data file using datahub (dcc) authentication, provide the dcc username instead of era-fasp and you will be prompted for the password.

```
ascp -QT -l 300m -P33001 \ 
dcc_name@fasp.sra.ebi.ac.uk:/vol1/fastq/ERR327/009/ERR3278169/ERR3278169_1.fastq.gz \ 
local/target/directory
```
Downloading Assembled and Annotated Sequence Data

Files in public FTP folders can also be downloaded using Aspera.

e.g. a WGS sequence set like ftp://ftp.ebi.ac.uk/pub/databases/ena/wgs/public/wy/WYAA01.dat.gz

```
ascp -QT -l 300m -P33001 -i path/to/aspera/installation/asperaweb_id_dsa.openssh /fasp-ebi@fasp.ebi.ac.uk:databases/ena/wgs/public/wy/WYAA01.dat.gz local/target/ → directory
```

9.3 Programmatic Access

When working with a large number of records or when developing an automated pipeline, it can be preferable to explore and interact with the programmatic services that ENA has to offer.

Once you are familiar with how ENA records are linked and what data are available associated with each record, please explore our more advanced guides for accessing data from the archive programatically:

9.3.1 How to Access ENA Programatically

There are a number of REST APIs available for programmatic access of the European Nucleotide Archive. These enable programmatic access to the functionality of the ENA Advanced Search as well as direct download of ENA records and associated files.

Please see the relevant guides below for examples and tutorials on ENA programmatic data access and retrieval.

Perform Searches

All functionalities of the ENA Advanced Search can be performed programmatically using a combination of the ENA Portal API and the ENA Browser API. You can download the API docs for the Portal API here and the Browser API here.

You can further explore related records outside of the European Nucleotide Archive by programmatically accessing the ENA Cross Reference Service.

For examples and tutorials on how to use these APIs, please see the guidelines below:

How to Perform Advanced Searches Across ENA Programatically

The ENA Advanced Search is a powerful tool for exploring the datasets within the archive using tailored search queries. The ENA Advanced search in the browser provides a user interface to build search queries and can be a useful way to start exploring the full functionality of the service. Once you are happy with a query, you can click the ‘Copy Curl Request’ button to allow you to run this locally on the command line.

This guide will help you explore the search functionalities of the ENA retrieval APIs. When first exploring the APIs, it can be worth testing out the query builder in the ENA Advanced Search interface as well as playing around with the APIs’ swagger interfaces (ENA Portal API and ENA Browser API). Using a combination of all three of these can help build more complex searches.

In this example we are going to perform a complex search and retrieve raw read and primary metagenome datasets specifically for cow rumen samples collected in the UK.

For a simpler overview on the advanced search, see our guide on using taxonomy to perform searches programmatically here.
Choosing the Right Data Type Result

Before performing a search, you will first need to determine what ‘result’ data type to search against. You can use the ENA Portal API to output the full list:

https://www.ebi.ac.uk/ena/portal/api/results?dataPortal=ena

<table>
<thead>
<tr>
<th>resultId</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>analysis_study</td>
<td>Studies used for nucleotide sequence analyses from reads</td>
</tr>
<tr>
<td>analysis</td>
<td>Nucleotide sequence analyses from reads</td>
</tr>
<tr>
<td>assembly</td>
<td>Genome assemblies</td>
</tr>
<tr>
<td>coding_release</td>
<td>Protein coding sequences (Release)</td>
</tr>
<tr>
<td>coding_update</td>
<td>Protein coding sequences (Update)</td>
</tr>
<tr>
<td>wgs_set</td>
<td>Genome assembly contig sets (WGS)</td>
</tr>
<tr>
<td>tsa_set</td>
<td>Transcriptome assembly contig sets (TSA)</td>
</tr>
<tr>
<td>environmental</td>
<td>Environmental samples</td>
</tr>
<tr>
<td>noncoding_release</td>
<td>Non-coding sequences (Release)</td>
</tr>
<tr>
<td>noncoding_update</td>
<td>Non-coding sequences (Update)</td>
</tr>
<tr>
<td>noncoding</td>
<td>Non-coding sequences</td>
</tr>
<tr>
<td>read_study</td>
<td>Studies used for raw reads</td>
</tr>
<tr>
<td>read_experiment</td>
<td>Experiments used for raw reads</td>
</tr>
<tr>
<td>read_run</td>
<td>Raw reads</td>
</tr>
<tr>
<td>sample</td>
<td>Samples</td>
</tr>
<tr>
<td>sequence_release</td>
<td>Nucleotide sequences (Release)</td>
</tr>
<tr>
<td>sequence_update</td>
<td>Nucleotide sequences (Update)</td>
</tr>
<tr>
<td>sequence</td>
<td>Nucleotide sequences</td>
</tr>
<tr>
<td>study</td>
<td>Studies</td>
</tr>
<tr>
<td>taxon</td>
<td>Taxonomic classification</td>
</tr>
</tbody>
</table>

From this list we can see that the read_run result provides a platform to search for ‘Raw reads’ and the analysis result provides a platform to search for ‘Nucleotide sequence analyses from reads’.

Why are we not using the assembly result?

You may be wondering why we are not using the assembly result to search for the primary metagenomes. Within ENA, the Genome Assembly database only accepts high-quality, individual-genome level assemblies such as isolate genomes, MAGs and SAGs. As primary metagenomes are sets of contigs with a mixture of genomic data from many individuals within an environmental sample, primary metagenome assemblies are archived as ‘sequence assembly’ type analyses.

Breaking Down Your Query

To build queries for the API, you need to string together individual search fields with AND or OR operators. This means you’ll need to break down your search question into individual search components.

In this example, the datasets we want to retrieve are those from ‘cow rumen samples collected in the UK’. So let’s break this query down into bite-size chunks. We want to retrieve data from:

samples collected from Rumen
AND only those of a Cow
AND only those collected in the UK
What Fields Can I use in My Search?

The easiest way to explore the fields you can use for your search is the ENA Advanced Search query builder but you can also output the full list in the Portal API by looking up the searchFields for each result:

```
https://www.ebi.ac.uk/ena/portal/api/searchFields?result=read_run
https://www.ebi.ac.uk/ena/portal/api/searchFields?result=analysis
```

Here, we can see in both searches, the following fields can be used:

- host_body_site - the site in the host organism where the sample was collected
- host_tax_id - the taxon ID for the host - this is the best way to accurately narrow down the host species sampled
- country - the country where the sample was collected

The tax ID for Bos Taurus is 9913. So, when we put these together as a query we get:

```
query=country="United Kingdom" AND host_tax_id=9913 AND host_body_site="rumen"
```

Why are we using host_tax_id instead of scientific name?

Here, we are searching for samples of the microbiome within a host not samples of the host genome itself. Microbiome samples use biome-level metagenome taxonomy and should include the host taxonomy in the metadata like in this example. You can also perform searches using this biome-level taxonomy to find microbiome data. See here for our guide on performing taxonomy-based searches.

Search For Raw Reads

To return a list of all the raw reads for our designed query, we can perform a search in the Portal API against the read_run result:

```
https://www.ebi.ac.uk/ena/portal/api/search?result=read_run&query=country="United Kingdom" AND host_tax_id=9913 AND host_body_site="rumen"
```

You can also specify the result format as ‘&format=tsv’ or ‘&format=json’. TSV is the default.

When looking at this search, we can see that only the accession and description are returned.

What Other Fields Can I Return?

The easiest way to explore the fields you can return from your search is the ENA Advanced Search query builder but you can also output the full list in the Portal API by looking up the returnFields for your result:

```
https://www.ebi.ac.uk/ena/portal/api/returnFields?result=read_run
```

So, in this example we wanted to access the resulting raw reads for the search. We can see the following field can be returned:

- submitted_ftp - the FTP links for the submitted raw read files

So we can add this to our search:

```
https://www.ebi.ac.uk/ena/portal/api/search?result=read_run&query=country="United Kingdom" AND host_tax_id=9913 AND host_body_site="rumen"&fields=submitted_ftp
```
You could also add the metadata fields that we specifically searched for to check the search performed as we expected:

```plaintext
https://www.ebi.ac.uk/ena/portal/api/search?result=read_run&query=country="United Kingdom" AND host_tax_id=9913 AND host_body_site="rumen"&fields=host_body_site,host_tax_id,country,submitted_ftp
```

**Fetching Full XML Records From Search**

Now that we’ve designed our search using the Portal API, we can optionally also run it in the Browser API to download the full XML records resulting from the search:

```plaintext
https://www.ebi.ac.uk/ena/browser/api/xml/search?result=read_run&query=country="United Kingdom" AND host_tax_id=9913 AND host_body_site="rumen"
```

The XML records can provide details on additional metadata that isn’t indexed for search as well as cross-reference links.

**Search For Analyses**

To return a list of all the primary metagenome assemblies for our designed query, we can perform a search in the Portal API against the **analysis** result. However, as analyses can be all kinds of different types, we’ll need to narrow down the search even further here.

In this example we want to search for *primary metagenome assemblies* so we can add the following search components:

- **analysis_type** - the type of analysis record - for assemblies this is `SEQUENCE_ASSEMBLY`
- **assembly_type** - we specifically want to search for ‘primary metagenome’ assemblies

Let’s add these to our query and search against the **analysis** result:

```plaintext
https://www.ebi.ac.uk/ena/portal/api/search?result=analysis&query=country="United Kingdom" AND host_tax_id=9913 AND host_body_site="rumen" AND analysis_type="SEQUENCE_ASSEMBLY" AND assembly_type="primary metagenome"
```

When looking at this search, again, we can see that only the accession and description are returned.

**What Other Fields Can I Return?**

Let’s check out the **returnFields** for the **analysis** result data type:

```plaintext
https://www.ebi.ac.uk/ena/portal/api/returnFields?result=analysis
```

Again we can see the submitted file FTP location can be returned:

- **submitted_ftp** - the FTP links for the submitted assembly files

So we can add this to our search:

```plaintext
https://www.ebi.ac.uk/ena/portal/api/search?result=analysis&query=country="United Kingdom" AND host_tax_id=9913 AND host_body_site="rumen" AND analysis_type="SEQUENCE_ASSEMBLY" AND assembly_type="primary metagenome"&fields=submitted_ftp
```

As before, you could also add the details we searched for to check the search performed as we expected:
Fetching Full XML Records From Search

Now that we’ve designed our search using the Portal API, again, we can optionally also run it in the Browser API to download the full XML records resulting from the search:

https://www.ebi.ac.uk/ena/browser/api/xml/search?result=analysis&query=country="United Kingdom" AND host_tax_id=9913 AND host_body_site="rumen" AND analysis_type="SEQUENCE_ASSEMBLY" AND assembly_type="primary metagenome"

Like before, the XML records can provide details on additional metadata that isn’t indexed for search as well as cross-reference links.

How to Search for Cross References in ENA Programmatically

The ENA Xref service holds cross-references to a number of external data resources linked to ENA records. The update and frequency of each source is dependent on their own release cycle and/or internal processes, with ENA supporting updates as frequently as once a week.

These cross-references can be explored programmatically using the Xref API which is documented with a Swagger interface.

This guide is not extensive and is designed to introduce you to some example uses for the Xref API which can be used as a platform for you to explore the API and service further.

Most of these examples use the ‘tsv’ format result but this can be swapped for ‘json’ if that is preferable.

Display List of All Cross Reference Sources

To get a good overview of what is included in the cross-reference service. You can first access the full list of cross-reference ‘Sources’ registered with ENA. These sources are the external data resources which are linked to ENA records. You can use the following endpoints to do this:

https://www.ebi.ac.uk/ena/xref/rest/tsv/source
https://www.ebi.ac.uk/ena/xref/rest/json/source

First 10 resulting cross-reference Sources as a TSV:

<table>
<thead>
<tr>
<th>Source</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ArrayExpress</td>
<td>ArrayExpress experiment</td>
</tr>
<tr>
<td>BBCC/LMBP</td>
<td>BBCC/LMBP Plasmid Collection</td>
</tr>
<tr>
<td>BlobToolKit</td>
<td>BlobToolKit: Toolkit for genome assembly QC</td>
</tr>
<tr>
<td>CABRI</td>
<td>CABRI</td>
</tr>
<tr>
<td>CCAP</td>
<td>Culture Collection of Algae and Protozoa</td>
</tr>
<tr>
<td>Citation</td>
<td>Citation</td>
</tr>
<tr>
<td>COMPARE-RefGenome</td>
<td>Reference Genome as provided by COMPARE</td>
</tr>
<tr>
<td>dictyBase</td>
<td>Dictyostelid genomics</td>
</tr>
<tr>
<td>Ensembl</td>
<td>Genome (Ensembl)</td>
</tr>
</tbody>
</table>

(continues on next page)
First 2 resulting cross-reference Sources in JSON:

```
[{
  "Description": "ArrayExpress experiment",
  "HomePage": "https://www.ebi.ac.uk/arrayexpress/",
  "LastUpdated": "2020-01-11 05:01:04.066495",
  "Source": "ArrayExpress"
}, {
  "Description": "BCCM/LMBP Plasmid Collection",
  "HomePage": "http://www.genecorner.ugent.be/",
  "LastUpdated": "2017-06-03 18:19:30.764629",
  "Source": "BCCM/LMBP"
}, {
  "Description": "EuGene: a database of gene expression and gene regulatory network",
  "HomePage": "http://eugene.cis.hku.hk/",
  "LastUpdated": "2018-03-31 00:00:00",
  "Source": "EuGene"
}]
```

In the above example, the TSV provides more direct readability but the JSON format provides additional information on the date the cross-references for that Source were most recently updated. It can be worth exploring both the ‘tsv’ and ‘json’ endpoints available before deciding what is most useful for your particular use-case.

In addition to providing an overview of the cross-reference service, this endpoint is useful for determining the Source name for any sources you may want to explore further.

**Look up Cross References for a Source**

Once you have determined what Source you would like to search for, you can perform a search against this Source. For example, to fetch records that have a cross-reference registered with MGnify (EMBL-EBI’s metagenomic data analysis service), you could look up the following:

```
https://www.ebi.ac.uk/ena/xref/rest/tsv/search?source=MGnify&limit=100
```

This results in a tsv of all records that have a cross-reference with MGnify. The cross-reference service provides the url to the cross-reference source (in this case MGnify’s website) as well as the record in ENA.

This example is limited to 100 records. By using the ‘limit’ and ‘offset’ options, you can retrieve the data in batches. By default the limit is set to 100,000 records. You can set the limit to 0 to fetch all the records.

**Narrow Down a Search By Target Record Type**

You can narrow down cross-reference searches further to only return records of a certain Type. For example, you may want to search specifically for sample records which are linked to the MGnify service.

Firstly, you may want to determine what Targets are available and how they are named. To list the full list of Target options and Target names, you can use the following endpoint:

```
https://www.ebi.ac.uk/ena/xref/rest/tsv/target
```

Result:
Here we can see that Samples are determined by the target ‘sample’. Now, you can narrow down your previous search:

https://www.ebi.ac.uk/ena/xref/rest/tsv/search?source=MGnify&target=sample&limit=100

Look up Cross References for a Record

As opposed to looking for cross-references by the registered service, you may want to look up all cross-references for a particular ENA Record. To do this, you can also perform a cross-reference search using an INSDC accession:

https://www.ebi.ac.uk/ena/xref/rest/tsv/search?accession=AY772730

Result:

<table>
<thead>
<tr>
<th>Source</th>
<th>Source primary accession</th>
<th>Source secondary accession</th>
<th>Source primary accession</th>
<th>Source secondary accession</th>
<th>Source</th>
<th>Source primary accession</th>
<th>Source secondary accession</th>
<th>Source</th>
<th>Source primary accession</th>
<th>Source secondary accession</th>
<th>Source</th>
<th>Source primary accession</th>
<th>Source secondary accession</th>
</tr>
</thead>
<tbody>
<tr>
<td>EuropePMC</td>
<td>PMC1393082</td>
<td>16517856</td>
<td>EuropePMC</td>
<td>PMC1393082</td>
<td>16517856</td>
<td>EuropePMC</td>
<td>PMC1393082</td>
<td>16517856</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC1594604</td>
<td>16891526</td>
<td>EuropePMC</td>
<td>PMC1594604</td>
<td>16891526</td>
<td>EuropePMC</td>
<td>PMC1594604</td>
<td>16891526</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC2828081</td>
<td>17953089</td>
<td>EuropePMC</td>
<td>PMC2828081</td>
<td>17953089</td>
<td>EuropePMC</td>
<td>PMC2828081</td>
<td>17953089</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC2897520</td>
<td>20484606</td>
<td>EuropePMC</td>
<td>PMC2897520</td>
<td>20484606</td>
<td>EuropePMC</td>
<td>PMC2897520</td>
<td>20484606</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC2919043</td>
<td>20554772</td>
<td>EuropePMC</td>
<td>PMC2919043</td>
<td>20554772</td>
<td>EuropePMC</td>
<td>PMC2919043</td>
<td>20554772</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC3096948</td>
<td>21524296</td>
<td>EuropePMC</td>
<td>PMC3096948</td>
<td>21524296</td>
<td>EuropePMC</td>
<td>PMC3096948</td>
<td>21524296</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC3110387</td>
<td>21686127</td>
<td>EuropePMC</td>
<td>PMC3110387</td>
<td>21686127</td>
<td>EuropePMC</td>
<td>PMC3110387</td>
<td>21686127</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC3187498</td>
<td>21840454</td>
<td>EuropePMC</td>
<td>PMC3187498</td>
<td>21840454</td>
<td>EuropePMC</td>
<td>PMC3187498</td>
<td>21840454</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(continues on next page)
Expanding metadata

In some cases, the cross-reference registered may have additional metadata. For example, cross-references registered with the source COMPARE-RefGenome.

To view this, add “expanded=true”:

https://www.ebi.ac.uk/ena/xref/rest/tsv/search?source=COMPARE-RefGenome&accession=AY772730&expanded=true

Result:

<table>
<thead>
<tr>
<th>Source</th>
<th>Source primary accession</th>
<th>Source secondary accession</th>
<th>Source url</th>
<th>Target url</th>
<th>Target primary accession</th>
<th>Target secondary accession</th>
<th>Target species level</th>
<th>1st below-species level</th>
<th>2nd below-species level</th>
<th>3rd below-species level</th>
<th>Aggregated taxonomic name</th>
<th>genome</th>
</tr>
</thead>
<tbody>
<tr>
<td>EuropePMC</td>
<td>PMC3367634</td>
<td>16485473</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC3367634</td>
<td>16485473</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC3493335</td>
<td>22943503</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC3493335</td>
<td>22943503</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC3695492</td>
<td>23630317</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC3695492</td>
<td>23630317</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC4298492</td>
<td>24089606</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC4298492</td>
<td>24089606</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC5380889</td>
<td>28181902</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC5380889</td>
<td>28181902</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC5746213</td>
<td>29284004</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC5746213</td>
<td>29284004</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC5746213</td>
<td>29284004</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC5746213</td>
<td>29284004</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC5874246</td>
<td>29593246</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC5874246</td>
<td>29593246</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC5874246</td>
<td>29593246</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC5874246</td>
<td>29593246</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC5911914</td>
<td>25946552</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC5911914</td>
<td>25946552</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC6511519</td>
<td>30531093</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC6511519</td>
<td>30531093</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC6511519</td>
<td>30531093</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC6511519</td>
<td>30531093</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7011714</td>
<td>31483239</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7011714</td>
<td>31483239</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7011714</td>
<td>31483239</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7011714</td>
<td>31483239</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7160709</td>
<td>29992776</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7160709</td>
<td>29992776</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7160709</td>
<td>29992776</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7160709</td>
<td>29992776</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7160966</td>
<td>32322405</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7160966</td>
<td>32322405</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7160966</td>
<td>32322405</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7160966</td>
<td>32322405</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7165577</td>
<td>16629981</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7165577</td>
<td>16629981</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7165577</td>
<td>16629981</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7165577</td>
<td>16629981</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EuropePMC</td>
<td>PMC7165577</td>
<td>16629981</td>
<td><a href="http://europepmc.org/abstract/PMC/">http://europepmc.org/abstract/PMC/</a></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

9.3. Programmatic Access 207
Retrieve and Download Records

All public records within ENA are available to retrieve from the ENA Browser API so records can be programatically downloaded directly from the API. Associated files can be downloaded using FTP or Aspera protocol.

For a quick summary of metadata and file retrieval locations of records, you can use the ENA file reports.

For further simplicity, enaBrowserTools can be downloaded and run locally on the command line to fetch files associated with records by accession. It can also be used to bulk download records related to a specified Sample or Study.

For examples and tutorials on how to use the Browser API, file reports and enaBrowserTools, please see the guidelines below:

How to Download Records using the ENA Browser API

The ENA Browser API can be used to download ENA Records in EMBL flat file, fasta or XML format dependant on the record type.

<table>
<thead>
<tr>
<th>File Format</th>
<th>Available Record Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML</td>
<td>Study, Sample, Run, Experiment, Analysis, Taxon</td>
</tr>
<tr>
<td>EMBL Flat File (text)</td>
<td>Sequences (including coding and non-coding sequences), WGS sets, TSA sets</td>
</tr>
<tr>
<td>Fasta</td>
<td>Sequences</td>
</tr>
</tbody>
</table>

API End-points

The Browser API has three endpoints for each file format available in the table above. Using the XML file format as an example, below is a summary of how to use these end-points to retrieve data.

XML Records contain metadata provided by the submitter as well as injected cross-reference links to other related resources and FTP locations of associated files.

The three endpoints available to access XML records in the Browser API are as follows:

**Fetch Records by Accession**

https://www.ebi.ac.uk/ena/browser/api/xml/<accession>

This allows direct access to the XML of a provided accession. The XML file will not be downloaded automatically.
Fetch Records by Free-Text Search

https://www.ebi.ac.uk/ena/browser/api/xml/textsearch/<query>

This will return all records in XML format which match your text search. It can be used to perform very broad searches and is great to quickly access records of a particular datatype. However, be aware that text searches can lead to false positives so for extensive complex and comprehensive searches, it is best to perform an advanced search.

Fetch Records Using an Advanced Search

https://www.ebi.ac.uk/ena/browser/api/xml/search/<query>

This allows you to perform an advanced search and directly download the resulting records as XMLs. This is most useful when used alongside the Portal API. See our guide on how to perform programmatic advanced searches here.

Fetch Records Linked to a Sample, Taxon or Study

https://www.ebi.ac.uk/ena/browser/api/xml/links/<data-type>

This allows you to directly download records associated with a specified Sample, Study or Taxon. You just need to specify which one of the three datatypes you want to download records from (sample, study or taxon) as well as the accession then specify which datatype you would like to download that is related to that record.

Retrieving ENA File reports

While file reports can be built using the ENA advanced search, there is a special filereport URL that will return summarised reports about a provided accession. This URL bypasses the advanced search and fetches information directly from a data cache, increasing the speed of delivery.

This faster resource is available using the following URL template:

http://www.ebi.ac.uk/ena/portal/api/filereport?accession=<accession>&result=<result>[
  →fields=<fields>]

When the fields parameter is used, only those columns requested are returned and they are in the same order as requested. If the fields parameter is not used, all available columns for the result are returned.

Example with all report fields:

http://www.ebi.ac.uk/ena/portal/api/filereport?accession=SRX017289&result=read_run

Example with specified return fields:

http://www.ebi.ac.uk/ena/portal/api/filereport?accession=SRX017289&result=read_run&
  →fields=run_accession,fastq_ftp,fastq_md5,fastq_bytes

Run Reports

To retrieve the run file report, the result should be set to read_run. Study accessions (ERP, SRP, DRP, PRJ prefixes), experiment accessions (ERX, SRX, DRX prefixes), sample accessions (ERS, SRS, DRS, SAM prefixes) and run accessions (ERR, SRR, DRR prefixes) can be supplied.
Analysis Reports

To retrieve the analysis file report, the result should be set to analysis. Study accessions (ERP, SRP, DRP, PRJ prefixes), sample accessions (ERS, SRS, DRS, SAM prefixes) and analysis accessions (ERZ, SRZ, DRZ prefixes) can be supplied.

How to Download Data files from ENA using enaBrowserTools

enaBrowserTools is a set of scripts that interface with the ENA web services to download data from ENA easily, without any knowledge of scripting required.

You will need to have Python installed and have a basic understanding on how to use the Command Line.

Download enaBrowserTools and have a look through the README for guides on installation, set up and general usage. The tutorials below will walk you through some examples of how to use enaBrowserTools to download data from ENA.

enaDataGet

To download all data for a given accession, use the command “enaDataGet”. This works for data-holding accessions such as Sequence records, Assembly records, Run or Analysis records or WGS sets.

Begin by printing the command’s “help” to the screen so you can see what arguments are available for enaDataGet:

```
> enaDataGet --help
```

Review the help until you are happy with what the command does and how to use it.

Now, let’s try an example and download the data files that are in the sequencing run ERR164409.

We can see from the help output that raw reads are available in ‘submitted’, ‘fastq’ and ‘sra’ formats:

- submitted - the file format provided by the submitter when they provided their data to the archive
- fastq - the standardised ENA fastq format (where the sequence names in the file are standardised)
- sra - a file format that works with the NCBI tool SRAToolkit.

Simply choose the desired file format, specify the accession and provide a destination directory and the raw read file will be downloaded:

```
> enaDataGet -f fastq -d <destination/directory> ERR164409
```

Unless you provide a destination directory, the data will be downloaded to the directory from which you run the command so it’s advised to use a ‘-d’ argument in your command.

In this case, the run ERR164409 has a library layout of SINGLE. If the sequencing data was paired-end then this command would download both files.

Including Metadata

If you wanted to also download the Run metadata, you could do this by including the “-m” argument which will also download the ENA Record in XML format:

```
> enaDataGet -f fastq -d <destination/directory> -m ERR164409
```

Note that the XML metadata download option is only available for Run and Analysis records.
enaGroupGet

The command “enaGroupGet” allows download of all data files of a certain type of Record that are associated with a Study, Sample or given Taxon. This means you can download all raw reads in a Study or all analyses of a Sample etc. in bulk.

Begin by printing the command’s “help” to the screen so you can see what arguments are available for enaGroupGet:

```bash
> enaGroupGet --help
```

Review the help until you are happy with what the command does and how to use it.

Now, let’s try an example and download all the raw read files associated with the Sample SAMEA2591084.

Again, we know that raw reads are available in ‘submitted’, ‘fastq’ and ‘sra’ formats so let’s try and download the submitted files this time. Please note that if data was submitted via one of our INSDC partners (NCBI or DDBJ) as opposed to through ENA, these submitted files will not be available.

```bash
> enaGroupGet -f submitted -d <destination/directory> SAMEA2591084
```

In this case, 6 files should be downloaded.

Including Metadata

If you wanted to also download the metadata for all the resulting Runs, you could do this again by including the “-m” argument:

```bash
> enaGroupGet -f submitted -d <destination/directory> -m SAMEA2591084
```

Explore Taxonomy and Related Records

All sample records in ENA have taxonomic assignment. As a result, the majority of records stored within the archive can be searched based on their taxonomy.

The ENA has a REST API for access to taxonomic information (e.g. lineage and rank) so taxonomic records can be explored programmatically. You can also download taxon records in XML format using the ENA Browser API or explore related records with the ENA Portal API.

For examples and tutorials on how to use the ENA taxonomy services, please see the guidelines below:

Programmatically Accessing Taxonomic Information

Every ENA sample object should have a taxonomic classification. The INSDC maintains a database of all unique taxonomic classifications known to us. Each classification has a unique id and this is expanded to show the scientific name and common name of the organism when the sample is viewed in the ENA Browser.

There are several ways that you can access information about taxonomy or related records using the REST APIs available through the ENA taxonomy services.

Finding General Information about Taxa

If you know the scientific name of the organism you can find the tax ID with this endpoint:
Simply append the scientific name to the URL:

```
> curl "https://www.ebi.ac.uk/ena/taxonomy/rest/scientific-name/Leptonycteris nivalis"
{
  "taxId": "59456",
  "scientificName": "Leptonycteris nivalis",
  "commonName": "Mexican long-nosed bat",
  "formalName": "true",
  "rank": "species",
  "division": "MAM",
  "lineage": "Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Mammalia; Eutheria; Laurasiatheria; Chiroptera; Microchiroptera; Phyllostomidae; Glossophaginae; Leptonycteris;",
  "geneticCode": "1",
  "mitochondrialGeneticCode": "2",
  "submittable": "true"
}
```

You can also do this with the common name using this endpoint:

```
www.ebi.ac.uk/ena/taxonomy/rest/any-name/
```

Simply append the common name to the URL:

```
> curl "https://www.ebi.ac.uk/ena/taxonomy/rest/any-name/golden%20arrow%20poison%20frog"
{
  "taxId": "377316",
  "scientificName": "Atelopus zeteki",
  "commonName": "golden arrow poison frog",
  "formalName": "true",
  "rank": "species",
  "division": "VRT",
  "lineage": "Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Amphibia; Batrachia; Anura; Neobatrachia; Hylidea; Bufonidae; Atelopus;",
  "geneticCode": "1",
  "mitochondrialGeneticCode": "2",
  "submittable": "true"
}
```

Please note that not all taxa have a common name.

If you do not know the scientific name or the common name but you have an idea, you can use the `suggest` endpoint:

```
www.ebi.ac.uk/ena/taxonomy/rest/suggest-for-submission/
```

See this example using the search term “curry”:

```
> curl "https://www.ebi.ac.uk/ena/taxonomy/rest/suggest-for-submission/curry"
(continues on next page)
```
Accessing Taxon XML Records and Full Lineage

You can access the XML record of any public taxon using the Browser API. For example, to access the record of the ant fungus garden metagenome taxon, we can provide the Browser API XML endpoint with the tax ID 797283.

https://www.ebi.ac.uk/ena/browser/api/xml/797283

You can also choose to download this directly from the API by specifying “download=true”:

https://www.ebi.ac.uk/ena/browser/api/xml/797283?download=true

This XML record provides general taxonomic information such as rank or translation genetic code as well as the scientific names and tax IDs of the parent and child taxa related to the record. This allows full exploration of the lineage of the taxon.

Finding Associated Records

For a report of all records associated with a taxon, you can use the Discovery Portal API. This API can provide a table of record counts as well as provide a list of record IDs and descriptions when provided a ‘result’ data type.

For example, to provide a report of all records that link to the ‘ant fungus garden metagenome’ taxon, we could provide the Portal API with the tax ID using the ‘links/taxon’ endpoint. This can be in tsv or json format:

https://www.ebi.ac.uk/ena/portal/api/links/taxon?accession=797283\&format=tsv

Result:

<table>
<thead>
<tr>
<th>result_id</th>
<th>description</th>
<th>entry_cnt</th>
<th>base_cnt</th>
<th>subtree_entry_cnt</th>
<th>subtree_base_cnt</th>
</tr>
</thead>
<tbody>
<tr>
<td>read_experiment</td>
<td>Experiment</td>
<td>236</td>
<td>12253983418</td>
<td>236</td>
<td>12253983418</td>
</tr>
<tr>
<td>sequence_update</td>
<td>Sequence (Update)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>sample</td>
<td>Sample</td>
<td>236</td>
<td>0</td>
<td>236</td>
<td>0</td>
</tr>
<tr>
<td>analysis_study</td>
<td>Study</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>analysis</td>
<td>Analysis</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>study</td>
<td>Study</td>
<td>15</td>
<td>0</td>
<td>15</td>
<td>0</td>
</tr>
<tr>
<td>assembly</td>
<td>Assembly</td>
<td>4</td>
<td>340564769</td>
<td>4</td>
<td>340564769</td>
</tr>
<tr>
<td>sequence_release</td>
<td>Sequence (Release)</td>
<td>10</td>
<td>2048</td>
<td>10</td>
<td>2048</td>
</tr>
<tr>
<td>wgs_set</td>
<td>Genome assembly contig set</td>
<td>4</td>
<td>0</td>
<td>4</td>
<td>0</td>
</tr>
<tr>
<td>noncoding_release</td>
<td>Non-coding (Release)</td>
<td>10</td>
<td>2048</td>
<td>10</td>
<td>2048</td>
</tr>
<tr>
<td>noncoding</td>
<td>Non-coding</td>
<td>10</td>
<td>2048</td>
<td>10</td>
<td>0</td>
</tr>
<tr>
<td>coding_update</td>
<td>Coding (Update)</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

(continues on next page)
From this summary, we can see that this taxon has 15 studies associated with it. To then see a report of the study IDs and descriptions, we can specify this with the addition of ‘result=study’:

https://www.ebi.ac.uk/ena/portal/api/links/taxon?accession=797283&format=tsv&result=study

Result:

<table>
<thead>
<tr>
<th>study_accession</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRJNA258031</td>
<td>Atta colombica refuse dump Targeted Locus (Loci)</td>
</tr>
<tr>
<td>PRJNA336974</td>
<td>Cyphomyrmex longiscapus fungus garden microbial communities from Gamboa,</td>
</tr>
<tr>
<td></td>
<td>→Panama metagenome</td>
</tr>
<tr>
<td>PRJNA336975</td>
<td>Apterostigma dentigerum fungus garden microbial communities from Gamboa,</td>
</tr>
<tr>
<td></td>
<td>→Panama metagenome</td>
</tr>
<tr>
<td>PRJNA336982</td>
<td>Leaf cutter ant microbial communities from the University of Wisconsin—</td>
</tr>
<tr>
<td></td>
<td>→Madison, USA, from External Dump - Dump Bottom metagenome</td>
</tr>
<tr>
<td>PRJNA336984</td>
<td>Leaf cutter ant microbial communities from the University of Wisconsin—</td>
</tr>
<tr>
<td></td>
<td>→Madison, USA, from External Dump - Dump Top metagenome</td>
</tr>
<tr>
<td>PRJNA336998</td>
<td>Leaf cutter ant microbial communities from the University of Wisconsin—</td>
</tr>
<tr>
<td></td>
<td>→Madison, USA, from fungus growing ant-garden - Acromyrmex fungus garden</td>
</tr>
<tr>
<td></td>
<td>→Combined metagenome</td>
</tr>
<tr>
<td>PRJNA337000</td>
<td>Leaf cutter ant microbial communities from the University of Wisconsin—</td>
</tr>
<tr>
<td></td>
<td>→Madison, USA, from fungus growing ant-garden - Atta cephalotes fungus garden</td>
</tr>
<tr>
<td></td>
<td>→Combined metagenome</td>
</tr>
<tr>
<td>PRJNA337001</td>
<td>Atta colombica fungus garden Top metagenome</td>
</tr>
<tr>
<td>PRJNA337002</td>
<td>Atta texana Internal Dump Top metagenome</td>
</tr>
<tr>
<td>PRJNA337003</td>
<td>Atta texana Internal Dump Bottom metagenome</td>
</tr>
<tr>
<td>PRJNA39805</td>
<td>Atta colombica Fungus Garden Metagenome</td>
</tr>
<tr>
<td>PRJNA62039</td>
<td>Atta cephalotes Fungus Garden Metagenome</td>
</tr>
<tr>
<td>PRJNA62041</td>
<td>Atta colombica Fungus Garden Top Metagenome</td>
</tr>
<tr>
<td>PRJNA62043</td>
<td>Atta colombica Fungus Garden Bottom Metagenome</td>
</tr>
</tbody>
</table>

When exploring links to taxon records, you can also specify a taxonomic node such as a genus or family rank taxon and request all links in that subtree. For example, if you would like a report of all studies associated with taxa that are under the tax node ecological metagenomes. You could specify this with the addition of “subtree=true”:

https://www.ebi.ac.uk/ena/portal/api/links/taxon?accession=410657&result=study&subtree=true

**Downloading Taxonomy Data via FTP**

Taxonomy data is available for bulk download through FTP at ftp://ftp.ebi.ac.uk/pub/databases/ena/taxonomy/.

<table>
<thead>
<tr>
<th>File</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>taxonomy.xml.gz</td>
<td>Full release of the taxonomy data in ENA taxonomy format.</td>
</tr>
<tr>
<td>sdwca</td>
<td>Full release of taxonomy data in Darwin Core Archive format.</td>
</tr>
</tbody>
</table>
GBIF and the Darwin Core Archive

The Global Biodiversity Information Facility (GBIF) aims to make the world’s Biodiversity data freely and universally available to provide an essential global informatics infrastructure for Biodiversity research and applications worldwide. Read about the Darwin Core Standard (DwC) on their website here.

The Darwin Core Archive comprises of 3 files: a tab-delimited data file, an XML file listing the descriptors of used in the data file and another XML file representing a metadata file with information related to the data itself, the data supplier, the archive creator name of the person who created the archive.

How to Programmatically Perform a Search Across ENA Based on Taxonomy

The Taxonomy of ENA Sample records and associated data are maintained in the NCBI taxonomy database. As a result, before searching through ENA using Taxonomy, you’ll need to find the appropriate taxon for the search within the Tax Database.

If you know the scientific name of the taxon for the search, you can retrieve the tax ID using the taxonomy database API:

```
www.ebi.ac.uk/ena/taxonomy/rest/scientific-name/<scientific-name>
```

If you do not know the scientific name or the common name but you have an idea, you can use the suggest endpoint:

```
www.ebi.ac.uk/ena/taxonomy/rest/suggest-for-submission/<search-term>
```

Search For All Reads of a Specified Taxon

As an example, let’s look at a search that will return all reads of the taxon ant fungus garden metagenome.

The ENA Discovery Portal API can be used to tailor powerful and complex searches across the archive. This API returns a report of the results of your search in TSV or JSON format. See the Swagger interface or API docs for full usage details.

For the purpose of a simple taxon based search, the query that would be run would use the parameter `tax_eq(<tax_id>)`.

The following search would provide all accessions and descriptions for raw reads which are from ant fungus garden samples (the default format is a tsv):

```
https://www.ebi.ac.uk/ena/portal/api/search?result=read_run&query=tax_eq(797283)
```

You can then further tailor your search to return particular fields of interest such as the sample collection date, the host sex of the sample, the sequencing platform etc. In this example, we will look at adding the following fields to the returned report:

- accession - the resulting Sample accession
- description - the record description
- collection_date - the sample collection date
- instrument_platform - the sequencing platform
- fastq_ftp - the FTP location of the fastq raw read files

You can view the Portal API docs for more information on what fields are available for different search types. These fields are added as a comma separated list to the API call:

```
https://www.ebi.ac.uk/ena/portal/api/search?result=read_run&query=tax_eq(797283),accession,description,collection_date,instrument_platform,fastq_ftp
```

9.3. Programmatic Access
Download the Resulting Sequencing Data Files

By returning the FTP location of the ftp files, you can now download the raw read files directly from the FTP.

Download the XML Records of the Resulting Reads

There are limitations to generating a metadata report from the ENA Portal API. The Portal API only holds indexed fields meaning any details or custom metadata that a submitter has provided will not be returned.

If you want to explore ENA records in full detail, you may wish to download the records themselves as full XMLs.
To do this, you can use the ENA Browser API and use the same search query there (without the specified fields to return). This will download all the resulting XML records for that particular taxon:

https://www.ebi.ac.uk/ena/browser/api/xml/search?result=read_run&query=tax_eq(797283)

Including The Taxon Sub-tree in Searches

If you would like to search at a higher level in the taxonomy tree, e.g. Genus level, then you can specify to return all records under that tax node. This allows you to capture all Records of a taxonomic lineage.

To do this, simply swap the tax_eq(<tax_id>) for tax_tree(<tax_id>).

For example, the search below searches for a summary of ALL metagenome raw read records by searching under the tax node metagenomes:

https://www.ebi.ac.uk/ena/portal/api/search?result=read_run&query=tax_tree(408169)

Sequence similarity search

EBI’s central NCBI BLAST service can be accessed via REST and SOAP.
For usage details and parameter options, see the NCBI BLAST+ documentation.

Access the CRAM Reference Registry

The CRAM reference registry provides access to reference sequences used in CRAM files. Retrieval of reference sequences from the CRAM reference registry is provided by MD5 or SHA1 checksum through the endpoints documented in the CRAM reference registry API.

CRAM Format

CRAM is a sequencing read file format that is highly space efficient by using reference-based compression of sequence data and offers both lossless and lossy modes of compression. The format specification for CRAM is maintained by the Global Alliance for Genomics and Health (GA4GH) whose members provide multiple implementations and coordinate future specification changes.

The CRAM reference registry is used by GA4GH Samtools.
CRAM Reference Registry reverse proxy

To reduce network traffic originating from the use of the CRAM Reference Registry we recommend using locally cached reference sequences. In addition to local caches supported by Samtools, it is possible to cache sequences using an HTTP proxy.

In the tutorial below, the Squid is used as a reverse proxy to cache reference sequences retrieved from the CRAM Reference Registry:

How to Cache CRAM Reference Sequences using Squid

To reduce network traffic originating from the use of the CRAM Reference Registry we recommend using locally cached reference sequences. In addition to local caches supported by Samtools it is possible to cache sequences using an HTTP proxy.

In the example below, the Squid is used as a reverse proxy to cache reference sequences retrieved from the CRAM Reference Registry.

Setting up Squid reverse proxy

1. Download and install Squid.

   If a binary package is not available for download then Squid can also be compiled from source. We recommend that you install version 3.1 or later.

2. Create directories for the cache files, log files and the Squid configuration file.

   In the example below we have created directories under /data/squid but any directory accessible to the user running Squid can be used after it has been defined in the Squid configuration file.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/data/squid/cache</td>
<td>Used to store the local squid cache</td>
</tr>
<tr>
<td>/data/squid/log</td>
<td>Used to store the Squid log files</td>
</tr>
<tr>
<td>/data/squid/conf</td>
<td>Used to store the Squid configuration file</td>
</tr>
</tbody>
</table>

3. Create the Squid configuration file.

   The following configuration file in /data/squid/conf will cache locally reference sequences retrieved from the CRAM Reference Registry. Parameter values that should be configured locally are bolded.

   ```
   max_filedesc 4096
   cache_mem <memory>
   maximum_object_size_in_memory 512 MB
   maximum_object_size 1024 MB
   http_port <port> accel defaultsite=<hostname> vhost
   #forwarded_for on
   cache_peer www.ebi.ac.uk parent 80 0 no-query originserver name=myAccel
   cache_dir ufs /data/squid/cache 65536 16 256
   acl mysites dstdomain ebi.ac.uk
   acl all src all
   http_access allow mysites
   http_access allow all
   cache_peer_access myAccel allow mysites
   cache_peer_access myAccel allow all
   access_log /data/squid/log/access.log
   ```
**Training Modules Documentation, Release 1**

(cache_store_log /data/squid/log/store.log)

(cache_log /data/squid/log/cache.log)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Example</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;port&gt;</td>
<td>8080</td>
<td>The HTTP port the Squid server is configured to use. Please note that ports below 1024 require root privileges.</td>
</tr>
<tr>
<td>&lt;hostname&gt;</td>
<td></td>
<td>Must match the hostname of the machine running the Squid server.</td>
</tr>
<tr>
<td>&lt;memory&gt;</td>
<td>10048 MB</td>
<td>The amount of memory reserved for Squid cache. It is recommended that this does not exceed 70% of the available physical memory. If the number of objects that is being cached exceeds the available memory then the cache will write the retrieved reference sequences to the Squid cache directory.</td>
</tr>
</tbody>
</table>

4. Initialize Squid.

Execute the following command:

```
squid -f /data/squid/conf/squid.conf -z
```

5. Start up Squid.

Execute the following command:

```
squid -f /data/squid/conf/squid.conf
```

This will start the squid process as a daemon so that it runs in the background. If you wish, you can execute squid in the foreground by adding the parameter -N.

6. Configuring Samtools to use Squid.

In Samtools, the \texttt{REF\_CACHE} environment variable is used to indicate that any downloaded reference sequences are stored locally in the specified directory in order to avoid subsequent downloads.

If you wish SAMTools to cache retrieved sequences in addition to caching them in Squid then you should define both \texttt{REF\_PATH} and \texttt{REF\_CACHE} environment variables:

where \texttt{<hostname>} and \texttt{<port>} are the ones used by the Squid server, and the \texttt{<dir>} is the directory path used by SAMTools for its cache. The \%2s/\%2s/\%s syntax informs SAMTools to store the sequences in sub-directories first using the first two digits and then using the second two digits of the MD5 checksum. This helps to avoid one large directory with thousands of files in it.

If you wish to cache reference sequences only in the Squid cache then only the \texttt{REF\_PATH} environment variable should be defined:
7. Configuring CRAMToolkit to use Squid.
CRAMTools can be instructed to use the Squid cache by defining the following Java property:

```
java -DREF_URL_TEMPLATE=http://<hostname>:<port>/ena/cram/md5/%s
```

where `<hostname>` and `<port>` are the ones used by the Squid server.

8. Testing Squid.
Execute the following command to test the cache:

The computed md5 checksum should match the requested entry. The cache status of the request can be determined by checking the `/data/squid/log/access.log` file. The cache status is available in the fourth column of the access log and can be one of the following:

To gracefully shut down the squid daemon process, execute the following command:

```
squid -k shutdown
```
Metadata objects can be updated done using either Interactive Webin or our Programmatic interface. Please be aware that not all attributes of objects can be updated, and any controlled vocabularies apply equally to updates as they do to the original submission.

10.1 Updating Studies, Samples, Experiments and Runs Interactively

The interactive submission interface supports some editing of your submitted objects. Access these existing objects by clicking the relevant tab after logging in to Webin.

Note that under no circumstances can an object’s own accession or alias attribute be edited.

When editing the XML version of an object, you should in general leave the element tags unchanged. These are the capitalised words enclosed in ‘<>’. For example, in the below XML snippet you should leave the words ‘ELEMENT’ unchanged and edit the ‘value’.

```xml
<ELEMENT>value</ELEMENT>
```

- 1: Study Edits
- 2: Sample Edits
- 3: Experiment And Run Edits

10.1.1 Study Edits

Some parts of the study object can be edited. These include the release date, title, description and publication references.
1. Login to Webin and find the studies tab.

2. Find the study in the list, or search for it by its accession.

3. If your study is confidential you can change the release date by clicking on the pencil icon and navigating to the required date in the calendar. To release the study simply select the current date/present day and set the following processes in motion:
   - Moving relevant sequence data files to a public archive.
   - Indexing and rendering the study and its objects so that they can be linked-to and visualised in the ENA browser.
   - Mirroring to INSDC databases, who will then make the data available through their services.

Please allow up to 48 hours for newly released data to appear in the public database. Read more about this in our Data Release Policies FAQ

4. For edits besides changing the release date, click the ‘Edit’ button.
5. The short name for the study will be visible in search outputs and overview pages whereas the descriptive title and abstract will be presented in the study’s public page.

6. You can add the PMID of any papers related to your data. There will then be a link to the paper from your study’s public page.

7. Study attributes are optional tag-value pairs you can specify to add extra information or to make your study more searchable. For example, you could add a ‘DOI’ tag with your paper’s DOI as the value.

8. Save your changes when you are satisfied with your updates, or click ‘Cancel’ to abandon them.

10.1.2 Sample Edits

To edit a sample, find it in the list (note the search box) and click the ‘Edit’ button next to it.

Your sample will be shown as an XML document which you can edit directly. Make changes as required and click the ‘Save’ button; your changes will not be saved if they invalidate the file. General XML errors and specific errors defined by us are prevented in this way. Note that not all fields can be edited: the sample alias and accession are immutable, and you will not be allowed to remove an attribute which is required by the specified checklist.
This method is useful for one-off edits but it is not feasible for editing many samples at once. For this you can use the programmatic method.
10.1.3 Experiment And Run Edits

Experiments and runs are both listed in the ‘Runs’ tab, where matched pairs of experiments and runs share a row in the table. Note that their are separate ‘Edit’ buttons for the two object types:

Be sure to use the correct edit button for the object you wish to edit. When you click the edit button, you will be shown the relevant object in XML format. Locate the element you wish to change and make the required changes, then click ‘Save’. You will not be able to save changes which invalidate the file.

Common Experiment Updates

The experiment object provides important metadata about how your data was produced. Common updates might include:

- Changing the library descriptor where a mistake has been made e.g. the library source could be listed as ‘GENOMIC’ when in fact it is ‘METAGENOMIC’
- Changing the &lt;SAMPLE_DESCRIPTOR&gt; where the experiment is pointing at the wrong sample
- Changing the &lt;STUDY_REF&gt; where the experiment is pointing a the wrong study
- Adding new &lt;EXPERIMENT_ATTRIBUTE&gt; elements to provide additional information about your experiment

All of the above can be achieved by editing the XML displayed when you click the ‘Edit’ button.

Common Run Updates

The most common run edit would be an MD5 update. You may need to do this if:

- An incorrect MD5 value has been registered for a file
- An invalid file has been replaced with a valid one, which has a different MD5 value

Find the &lt;FILE&gt; element’s ‘checksum’ attribute and correct the 32-digit value.

It is not possible to replace the uploaded file in this way; entering a new filename will not be accepted. If the submitted file has passed validation and been archived, it cannot be replaced. If the submitted file has failed validation, it must be replaced with an identically-named, corrected file.

10.2 Updating a Study Submission Programmatically

The main difference between programmatically creating new studies and updating existing ones is that the MODIFY action should be used instead of the ADD action:
Please note that the new study XML must either contain the original alias or the assigned accession number for the correct study to be updated.

Detailed advice on how to programmatically update studies using XML including instructions for retrieving previously submitted Study XMLs can be found in Programmatic Submission Options. Study XMLs can also be manually edited using the Webin submissions portal.

10.3 Updating a Sample Submission Programmatically

The main difference between programmatically creating new samples and updating existing ones is that the `MODIFY` action should be used instead of the `ADD` action:

```xml
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <MODIFY/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```

Please note that the new sample XML must either contain the original alias or the assigned accession number for the correct sample to be updated.

Detailed advice on how to programmatically update samples using XML including instructions for retrieving previously submitted Sample XMLs can be found in Programmatic Submission Options. Sample XMLs can also be manually edited using the Webin submissions portal.

10.4 Updating a Raw Read Submission Programmatically

The main difference between programmatically creating new experiment and runs and updating existing ones is that the `MODIFY` action should be used instead of the `ADD` action:

```xml
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      <MODIFY/>
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```

Please note that the new experiment or run XML must either contain the original alias or the assigned accession number for the correct object to be updated.

Detailed advice on how to programmatically update XMLs including instructions for retrieving previously submitted XMLs can be found in Programmatic Submission Options. Experiment and run XMLs can also be manually edited using the Webin submissions portal.
10.4.1 Reasons for Updating Experiment XML

Typical reasons for updating experiment XMLs include:

- Change the library descriptor:

```
<LIBRARY_DESCRIPTOR>
  <LIBRARY_NAME/>
  <LIBRARY_STRATEGY>RNA-Seq</LIBRARY_STRATEGY>
  <LIBRARY_SOURCE>TRANSCRIPTOMIC</LIBRARY_SOURCE>
  <LIBRARY_SELECTION>cDNA</LIBRARY_SELECTION>
  <LIBRARY_LAYOUT>
    <PAIRED NOMINAL_LENGTH="250" NOMINAL_SDEV="30"/>
  </LIBRARY_LAYOUT>
  <LIBRARY_CONSTRUCTION_PROTOCOL>Messenger RNA (mRNA) was isolated using the Dynabeads mRNA Purification Kit (Invitrogen, Carlsbad Ca. USA) and then sheared using divalent cations at 72°C. These cleaved RNA fragments were transcribed into first-strand cDNA using II Reverse Transcriptase (Invitrogen, Carlsbad Ca. USA) and N6 primer (IDT). The second-strand cDNA was subsequently synthesized using RNase H (Invitrogen, Carlsbad Ca. USA) and DNA polymerase I (Invitrogen, Shanghai China). The double-stranded cDNA then underwent end-repair, a single `A? base addition, adapter ligation, and size selection on anagarose gel (250 * 20 bp). At last, the product was indexed and PCR amplified to finalize the library prepration for the paired-end cDNA.</LIBRARY_CONSTRUCTION_PROTOCOL>
</LIBRARY_DESCRIPTOR>
```

- Move the experiment to a different study by changing the STUDY_REF:

```
<STUDY_REF accession="...">
</STUDY_REF>
```

- Associate the experiment with a different sample by changing the SAMPLE_DESCRIPTOR:

```
<SAMPLE_DESCRIPTOR accession="..">
</SAMPLE_DESCRIPTOR>
```

- Include more information in experiment attributes:

```
<EXPERIMENT_ATTRIBUTE>
  <TAG>library preparation date</TAG>
  <VALUE>2010-08</VALUE>
</EXPERIMENT_ATTRIBUTE>
```

10.4.2 Reasons for Updating Run XML

Most common reason for updating run XMLs is to change the file checksum:

```
<FILE filename="mantis_religiosa_R1.fastq.gz" filetype="fastq"
  checksum_method="MD5" checksum="9b8932f85ca54e687eba62fca3edce2"/>
```

You may need to change the checksum attribute for a file when a file has failed to be archived because of a mismatch between the provided and computed MD5 checksums.
Updates to assemblies can be performed through Webin-CLI, and result in the creation of a new version of the assembly.

Updates to assemblies are performed in very similar fashion to submission of assemblies. It is therefore sufficient to follow the guidance in the assembly submission pages, while considering the below factors. Note that this information applies only to updating the sequence and annotation, not the general assembly metadata.

**Study and sample reference must be maintained:** To submit an assembly update, make sure you reference the same study and sample accessions as were used in the original submission. In fact, this study-sample pair is unique to your assembly and is the means by which you submission is recognised as an update rather than a new assembly.

**Chromosome names must be maintained:** If you are updating a chromosome-level assembly, all chromosomes in the original assembly must be present in the updated version, with identical name. Webin-CLI does not currently validate for this and the error will only be caught after submission. If you believe you may have mistakenly submitted an update with a chromosome absent or incorrectly named, please contact our helpdesk.

The chromosome name is the value our pipeline uses to identify which sequence in your update should be considered the successor to each of the original chromosomes. The name is specified in the chromosome list file, as well as in the identifier lines of your FASTA or the ‘AC * ’ lines of your EMBL file.

Note that it is acceptable to include new chromosomes, as long as all previous chromosomes are maintained. Also, it is acceptable to update a contig or scaffold-level assembly to a chromosome-level assembly, in which case there are not yet any chromosome names to maintain.

**Annotation status must be maintained:** If your assembly was previously submitted with annotation and you are updating the assembly you can do this following the above factors, i.e. maintaining the study and sample, and adding an updated assembly including annotation.

If your assembly did not originally include annotation, it is not possible to update your assembly with annotation. An unannotated assembly can only be updated with new versions of the same unannotated assembly. In order to add annotation to a previously unannotated assembly you need to re-submit the assembly. Note this will result in new accession numbers. Please contact the helpdesk to request suppression of the existing assembly version.

**Use a new ASSEMBLYNAME:** The Webin-CLI manifest format includes an ‘ASSEMBLYNAME’ field. This must be unique in each of your submissions, whether they are updates or new assemblies. Using a pre-existing assembly name will result in an error informing you the object you are trying to submit already exists:
ERROR: In analysis, alias: "webin-genome-ASSEMBLYNAME", accession:"". The object being added already exists in the submission account with accession: "ERZxxxxxxx". The submission has failed because of a system error.

To check the name used for the previous submission, look up its ERZ accession in interactive Webin. Enter the XML view and find the ‘alias’ field, which will resemble “webin-genome-ASSEMBLYNAME”.

**Taxonomy changes require our assistance:** If there is a problem with the taxonomy of your sample, please contact our helpdesk. This process will require first that the sample taxonomy be correct, so you are welcome to update your sample first. If the taxon you wish to use does not exist in the taxonomy database, you will need to request its addition.

**Contact us for other metadata updates:** The advice given here pertains to updating the sequence and annotation (if applicable) of your assembly. If you wish to change the assembly metadata values, such as the coverage, please contact our helpdesk.

**State of publicity will be the same:** If the original assembly is already public, then the update will be made public as soon as processing is complete. However, please be aware that this process can take some time to complete, and you should allow at least a week after successful submission for your update to be apparent.

**Allow 30 minutes after original submission:** If the update is submitted less than 30 minutes after the original submission, it will fail after submission. This is a failsafe to prevent duplicate submissions, but does occasionally inconvenience some genuine use cases.

**Metagenomic Assemblies:** Primary and binned metagenome assemblies cannot be updated. If you wish to update one of these assemblies, please contact our helpdesk so that we can suppress the existing version. However, MAG assemblies work the same as isolate assemblies in the context of updates.
If you require updates to assembled and annotated sequences, please contact our helpdesk detailing this.
CHAPTER 13

Data Release Policies

13.1 How Is Data Release Managed?

Data release is controlled from the level of study objects. When a study is made public, all samples, experiments, runs and analyses associated with it are also made public. For an explanation of these terms and their relationships, please read the Metadata Model page.

13.2 Can I Advance/Postpone The Release Date?

You can edit the release date of a study from within your Webin account.

First, log into Interactive Webin. In the ‘Studies’ tab, find the study you wish to change and click the pencil icon adjacent to its current release date. You can move the date to be sooner or later as required. When making the date later, the new date cannot be more than 2 years beyond the current date.

When you are ready to release your study, you should do it by changing the release day to the current day.

Once the release date has elapsed, it will no longer be possible for you to affect the publicity of your data. See further down the page for information on this.

13.3 I Released My Study, Why Can’t I Find It In The Public Database?

After a study is made public, it can take up to 48 hours for your data to be searchable in the ENA Browser. It will take four days for it to be visible in GenBank. Please take this into consideration if you have a strict deadline by which your data must be public. Note that the processes which are taking place in this time are automated, and we are not
able to intervene in order to expedite the release of your data. If after 48 hours your data remain confidential, please contact our helpdesk.

### 13.4 Can I Make My Study Partially Available?

Whether you wish to make only parts of your submission available, or wish to make your submission available only to select users, this is unfortunately not a service we provide. Our policy is that submissions should be available in whole, to all users. If you need evidence of your submission for reviewers, forward them the email confirming your successful submission. If there are parts of your study which for some reason you no longer wish to release, please contact our helpdesk.

### 13.5 I Need To Remove My Confidential Study

If you have registered objects in the database which you wish to have removed, we can cancel them if they are not yet public. Contact our helpdesk with an explanation of the issue and we will handle the deletion for you. Please be very clear about which accessions should be removed, especially if you only want to have a few objects removed from the study.

### 13.6 I Need To Remove My Public Study

If your study is public and needs to be removed from public view on a permanent or temporary basis, you must contact our helpdesk and provide an explanation of the issue, along with the accession(s) to be suppressed. A curator will assess the request and carry it out if deemed suitable. Information about circumstances in which suppression will be considered can be found in our Data Availability Policy. Please note that data that is made public for even a short time may be used by downstream services which we do not control.

If your suppression will be temporary, please provide a new release date when you contact us.

### 13.7 Data Availability Policy

For full details of the INSDC data availability policy including the different levels of controlled access, see here.
Common Run Submission Errors

When you submit read data to ENA, we store and accession these as ‘Runs’. As part of the submission process, read data files must be uploaded to your Webin account’s FTP directory. After you complete the submission, several validation procedures are applied to the file(s). If validation is successful, files are archived, otherwise all account contacts are notified of the error(s).

The errors discussed here typically do not require you to repeat the submission in its entirety. It is usually sufficient to upload a corrected version of the file, possibly updating its MD5 value. Once you correct your submission, it can take 24 hours or longer for this to be registered and for the error notifications to cease.

The common error types are described here.

- **Error: Invalid File Checksum**
- **Error: Number Of Lines Is Not A Multiple Of Four**
- **Error: File Integrity Check Failed**
- **Error: Missing File**

A couple of general-purpose solutions are described too:

- **Appendix: Correcting An MD5 Value**
- **Appendix: Re-Uploading Your File**

If your problem is not described on this page, or you are not clear on the solution, please contact us through our support form.

### 14.1 Error: Invalid File Checksum

If this error occurs, you will receive an email containing something similar to the below:

<table>
<thead>
<tr>
<th>FILE_NAME</th>
<th>ERROR</th>
<th>MD5</th>
<th>FILE_SIZE</th>
<th>DATE</th>
<th>RUN_ID/ANALYSIS_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>mbr_depth_05.bam</td>
<td>Invalid file checksum</td>
<td>594934819a1571f805ff299807431da4</td>
<td>895557023</td>
<td>20-DEC-2016</td>
<td>ERR1766300</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>235</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
14.1.1 The Problem

The checksum is a means of checking a file has been uploaded in its entirety. It is a 32-character string calculated from the file, and is unique to that file. We recalculate the MD5sum after you complete your submission and confirm that it matches the value you registered. Therefore, if the upload procedure fails to deliver the full file, this will be evident from the checksum. You will have calculated this value previously and included it in your submission: you can see the value you registered in the notification email, as is the case above. If you used the graphical Webin File Uploader program, the MD5 will have been calculated automatically for you.

The error could indicate a failure in the file transfer process as described, or it could be that the wrong MD5 value was registered in the first place.

14.1.2 The Solution

Depending on the exact cause of the error, there are two possible solutions. Please see Appendix: Correcting An MD5 Value for information on how you can calculate the MD5 value of your local copy of the file and determine whether it matches the originally registered value. If they match, you are looking at a corrupted file error. If they do not match, you are dealing with an incorrectly registered value. In either case, please refer to the relevant section below.

Corrupted File: Upload Again

If you recalculate the MD5 value of you file locally and it matches the value you registered, it is likely the file upload was incomplete or corrupt. You therefore need to reupload the file. Once this is done, the file should be automatically accepted within 24 hours with no further action taken.

Please see Appendix: Re-Uploading Your File for information on how to replace the uploaded file.

Wrong MD5 Value Registered: Register a New One

If you recalculate the MD5 value of your local file and it does not match the value you registered, this will be the root of the problem.

You will need to re-register the MD5 value. Please see Appendix: Correcting An MD5 Value for information on how to do this.

If you have many runs to update, you may wish to do this programmatically, by submitting corrected XML versions of your runs. View our pages on Programmatic Run Updates to learn more about this.

14.2 Error: Number Of Lines Is Not A Multiple Of Four

You will receive an email resembling the below if this error occurs:

<table>
<thead>
<tr>
<th>FILE_NAME</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>soc9/MCONS1_R1.fq.gz</td>
<td>File content missing or malformed, Number of lines in fastq is not multiple of 4</td>
</tr>
<tr>
<td></td>
<td>c2f8455c1a024cfb96a6c91f5d71f534</td>
</tr>
</tbody>
</table>
14.2.1 The Problem

This error is specific to FASTQ files: each entry in such a file should comprise exactly four lines, none of which should be blank. The FASTQ format is as follows:

```
@<run accession>.<spot index> [_<spot name>_][<read index>]
<bases>
+
<phred qualities, ASCII encoded starting with '!'
```

If your file does not match this format, it may be incorrectly formatted, or may have become corrupted in the upload process.

14.2.2 The Solution

You can replicate the check we run on your file locally from the command line:

```
$ zcat MCONS1_R1.fq.gz | grep '^[[:space:]]' | wc -l
```

This command will output the amount of lines in the file, after removing any blank lines. Please note that blank lines are considered a violation of the FASTQ format, as empty reads are not informative.

If the line is count divisible by four, it is likely the file was corrupted during upload and should be reuploaded. If it is not divisible by four, you should discover why, correct your file and reupload.

Note: If you reformat your file and then reupload it, you will also need to re-register the checksum. See the Appendix: Correcting An MD5 Value for information on how to do this.

14.3 Error: File Integrity Check Failed

You will know this error has occurred if you receive an email resembling the below:

<table>
<thead>
<tr>
<th>FILE_NAME</th>
<th>ERROR</th>
</tr>
</thead>
<tbody>
<tr>
<td>UK/BR1-20_2.fq.gz</td>
<td>File integrity check failed, Can't unzip file</td>
</tr>
<tr>
<td></td>
<td>ef7e73ed95f64355d7bf7d48636b704f</td>
</tr>
<tr>
<td>cebiorepl.bam</td>
<td>File integrity check failed, File cannot be read using samtools</td>
</tr>
<tr>
<td></td>
<td>cecfa479356456cb6770986a6141bc44</td>
</tr>
<tr>
<td>frger.cram</td>
<td>File integrity check failed, Can't count number of records in the</td>
</tr>
<tr>
<td></td>
<td>807a0f6ida013916clca5f609b42526</td>
</tr>
</tbody>
</table>

14.3.1 The Problem

Submitted files are checked to confirm they can be unpacked. The specifics of how this is done depends on the file type, e.g. gzipped FASTQ files are checked with gunzip, while BAM files are checked with SAMtools.

Unpacking will fail for one of two reasons: either the uploaded file was corrupt to begin with, or the upload procedure did not complete fully and a corrupted file was received.
14.3.2 The Solution

You should start by confirming the integrity of your local copy of the file. Find instruction on how to do this by referring to the relevant subsection for your file type below.

If the file unpacks correctly, most likely corruption occurred during upload: reupload it to your submission directory. See Appendix: Re-Uploading Your File for information on how to do this.

If the file is identified as having errors, remake the file and upload this. Be sure to check that the MD5 value hasn’t changed: you will need to re-register a correct value if it has.

**FASTQ Files**

Our pipeline performs the following check on gzipped FASTQ files to validate them:

```bash
$ zcat BR1-20_2.fq.gz > /dev/null 2>&1
$ echo $? 
```

This will attempt to read the content of the file and print an exit code. If this value is 1 or higher, there is a problem with the file. You can try this on your local file to check its validity, then upload a corrected version. See Appendix: Re-Uploading Your File for information on how to do this.

**BAM Files**

The check performed on BAM files is as follows:

```bash
$ samtools view cetbiorepl.bam > /dev/null 2>&1 
$ echo $? 
```

This command attempts to view the BAM file and output the exit code of this procedure. If the code is 1 or higher, there is a problem with the file. Try this check on your local file and then upload a corrected version. See Appendix: Re-Uploading Your File for information on how to do this.

14.4 Error: Missing File

If a missing file error occurs, you will receive the below message:

<table>
<thead>
<tr>
<th>FILE_NAME</th>
<th>ERROR</th>
<th>MD5</th>
<th>FILE_SIZE</th>
<th>DATE</th>
<th>RUN_ID/ANALYSIS_ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>UFMG-CM-Y030_R1.fastq.gz</td>
<td>Missing file</td>
<td>2da9b9c9bb8833c14b103e0de123829c</td>
<td>137298909</td>
<td>13-JUN-2020</td>
<td>ERR2299965</td>
</tr>
</tbody>
</table>

14.4.1 The Problem

Submitted files occasionally go missing and must either be replaced or resubmitted.

14.4.2 The Solution

You should reupload the file to your submission area. Note that if you uploaded the original file to a subdirectory in your submission area, you must also upload the new file to this subdirectory. The processing pipeline expects to see the file for your run in the originally specified location, so this must be maintained. You can check what path the
pipeline is expecting to see by referring to the ‘FILE_NAME’ field of the error message: this will contain the full path. See Appendix: Re-Uploading Your File for information on how to correctly upload your file.

14.5 Appendix: Correcting An MD5 Value

If the MD5 value registered for your read file is incorrect, you can supply a corrected version. To do this:

1. Log into Interactive Webin
2. Switch to the ‘Runs’ tab
3. Enter the accession of the errored run into the search box
4. Click the ‘Edit’ button for the relevant run (there will be two edit boxes, use the rightmost one)
5. Enter the new MD5 value next to the correct file
6. Click ‘Save’

The change you have made will require 24 hours to take effect.

Calculating the MD5 value you need can be done natively from the command line in Mac/Linux. One of the following commands will work, if you supply the correct filename:

```
$ md5sum mbr_depth_05.bam
594934819a1571f805ff299807431da4 mbr_depth_05.bam

$ md5 mbr_depth_05.bam
594934819a1571f805ff299807431da4 mbr_depth_05.bam
```

Windows users should see the Microsoft Support Article on this subject.

14.6 Appendix: Re-Uploading Your File

If your error requires a new version of the file be uploaded, you have two options for this. You should first consider whether your file was originally uploaded to a sub-directory. You can tell by referring to the original error message, looking out for the ‘FILE_NAME’ column. The below error describes a file which was uploaded to a subdirectory:

```
SOC9/MCONS1_R1.fq.gz | File content missing or malformed, Number of lines in fastq is not multiple of 4 | c2f8455c1a024cfb96a6c91f5d71f534 | 1358349886 | 01-DEC-2016 03:12:35 | ERR1755094
```

You can tell this was uploaded to a subdirectory because the actual filename ( MCONS1_R1.fq.gz ) is preceded by a directory name and a ‘/’ character ( SOC9/ ). The replacement file must be uploaded to this same subdirectory, as this is where the processing pipeline expects to find it. Having determined this, refer to the relevant section below.

In either case, you may need to update the MD5 value if the originally registered value was correct for the originally uploaded file. If you need to update the MD5 value, please refer to Appendix: Correcting An MD5 Value.

14.6.1 If Your File Is Not In A Subdirectory

Please view our guidance on the Webin File Uploader. This will conveniently allow you to upload your file to the top level of your submission directory.
14.6.2 If Your File Is In A Subdirectory

You will need to upload your file using FTP Client. There are various options for doing this, described at the linked page.

If using a command line solution: Once you are connected to the FTP server, use the `ls` command to view the content of the directory and the `cd <directory-name>` command to move into the required location. Once you arrive in the desired directory, proceed to upload the files.
Samples are objects in the ENA database which describe biological material used in sequencing experiments. Common errors experienced when attempting to submit samples are discussed here.

15.1 Error: Object Being Added Already Exists In Submission Account

This error indicates that a sample alias in your submission has previously been used. The sample alias must uniquely identify the sample among all samples submitted in your Webin account. Locate the alias in line 3 of this Sample XML: ERS2513358.

During interactive submissions, the alias is referred to as the ‘Unique Name’: if nothing is entered here, samples are automatically given numerical values starting at 1. This means that if the field is ignored on two separate submissions, the second submission will result in this error.

![Basic Details](image)

Enter an informative name here, as you may need to use the alias in the future and it cannot be changed after submission.

Note that if you register a sample and then have it deleted, the alias still cannot be reused.
The classification system for source biological organisms for all INSDC records is the NCBI Taxonomy and is available from the ENA browser. The ENA team work alongside taxonomists at the NCBI to ensure that all ENA records display the accepted organism name and classification hierarchy. NCBI Taxonomy covers the complete tree of life and also includes other types, such as synthetic constructs and environmental samples. However, it is an incomplete classification system in that it only considers taxa for data that are represented in INSDC records. Users should note that taxa are only displayed if at least one associated ENA record is available.

### 16.1 Choosing the Right Taxonomy For Your Submission

Submitted organism names must be at ‘species’ rank. This rank type does not automatically mean the name is a published binomen; it is simply a rank, which differentiates the sequenced organism from another. For example, unidentified strains of the same bacterial genus should be kept as separate species, rather than binned together under the same genus name.

The interactive submission service has a look up table which you can use to find appropriate taxonomic identifiers.

Programmatic submitters will apply the taxonomic information to the sample object using the sample_name block:
If you do not know the scientific name or the common name that you would like to use for your submission but you have an idea, you can use this suggest endpoint for the ENA taxonomy service:

www.ebi.ac.uk/ena/taxonomy/rest/suggest-for-submission/

For example, using curl or pasting the URL in the browser for “curry” looks as follows:

```bash
> curl "http://www.ebi.ac.uk/ena/taxonomy/rest/suggest-for-submission/curry"
[
  {
    "taxId": "159030",
    "scientificName": "Murraya koenigii",
    "displayName": "curry leaf"
  },
  {
    "taxId": "261786",
    "scientificName": "Helichrysum italicum",
    "displayName": "curry plant"
  }
]
```

### 16.1.1 Checking a taxon is submittable

If you know the taxon you would like to use, you can check if its submittable and find any additional information about it by using one of the following urls:

www.ebi.ac.uk/ena/taxonomy/rest/taxon/scientific-name/

www.ebi.ac.uk/ena/taxonomy/rest/taxon/any-name/

www.ebi.ac.uk/ena/taxonomy/rest/tax-id/

For example, using curl or pasting the url in the browser for “mixed culture” looks as follows:

```bash
> curl "https://www.ebi.ac.uk/ena/taxonomy/rest/scientific-name/mixed%20culture"
[
  {
    "taxId": "1306155",
    "scientificName": "mixed culture",
    "formalName": "false",
    "rank": "no rank",
    "division": "UNC",
    "lineage": "unclassified sequences; ",
    "geneticCode": "1",
    "mitochondrialGeneticCode": "2",
    "plastIdGeneticCode": "11",
    "submittable": "false"
  }
]
```
Please see our guide on exploring taxonomy for more advice on exploring our taxonomy services programatically.

16.2 Environmental Taxonomic Classifications

16.2.1 Environmental Biome-Level Taxonomy

Every sample object in ENA must have a taxonomic classification assigned to it. Environmental samples can not be described with a single organism identifier because they represent an environment with an unknown variety and number of organisms.

For this purpose there are entries in the Tax Database that apply exclusively to environmental samples. Taxa of this type can be immediately identified as they contain the term “metagenome” as part of the scientific name. These are searchable within the Tax Database using the same methods described above.

```
curl "https://www.ebi.ac.uk/ena/taxonomy/rest/suggest-for-submission/marsupial%20meta"
{
  "taxId": "1477400",
  "scientificName": "marsupial metagenome",
  "displayName": "marsupial metagenome"
}
```

To view all environmental taxonomy available please visit the “metagenomes” tax node. Click on the Tax tree tab and click the arrow icons to expand the categories:

The metagenomic term that is used to describe the biome is also the scientific name of the chosen taxon and can be used to find the tax ID in the same methods described above. For example, you can find the tax ID for *termite fungus garden metagenome* here:

```
www.ebi.ac.uk/ena/taxonomy/rest/scientific-name/termite_fungus_garden_metagenome
```

Please note that new metagenome taxonomic records are rarely added, particularly those that add granularity. Please use the closest available choice, even if this is a less granular option. Only request a new term if you are sure you are unable to use anything in the lists available.
16.2.2 Environmental Organism-Level Taxonomy

If you are submitting sequences or assemblies that have been identified taxonomically from homology alone, with no culturing or isolation of the organism beforehand, then we also consider these as environmental samples. These samples should be registered with uncultured taxonomy to make it clear they were derived from an environmental source.

A typical use-case of this would be the submission of a single fully assembled genome from a mixed DNA sample (i.e., from a metagenomic source). There are exceptions where this taxonomy should not be used: for example, organisms which can be reliably recovered from their diseased host (e.g., endosymbionts, phytoplasmas) and organisms from samples which are readily identifiable by other means (e.g., cyanobacteria); organisms such as these are not considered uncultured/environmental in the way described here. If you are unsure whether your sample should be registered as environmental, contact our helpdesk for assistance.

Environmental organism-level taxa are usually prefixed with the term uncultured and are not allowed to have a species epithet. Some examples of basic organism names that can be used include:

- uncultured bacterium (taxid:77133)
- uncultured archaeon (taxid:115547)
- uncultured cyanobacterium (taxid:1211)
- uncultured prokaryote (taxid:198431)
- uncultured fungus (taxid:175245)
- uncultured eukaryote (taxid:100272)

Where possible, you should register your samples with the most granular identification possible, up to Genus level. For example, for prokaryotes, the format is:

uncultured <Rank> sp.

* e.g. uncultured Bacillus sp.

For Fungi, the ‘sp.’ is dropped:

uncultured <Rank>

* e.g., uncultured Glomus

When registering samples that use uncultured taxonomy, a general environmental record should also be registered to describe the biome that was originally sequenced. This biome-level environmental sample should also be referenced within the organism-level sample using the “sample derived from” attribute.
Requesting New Taxon IDs

All submissions to INSDC databases must be classified against the NCBI Taxonomy database. Entries in the Taxonomy database have integer taxon IDs and are visible in the ENA browser alongside their full lineage and available data (e.g. human Tax ID is 9606). This database covers the complete tree of life, as well as viruses, synthetic constructs and metagenomic taxa. However, it is incomplete in that it only considers taxa which are already represented in INSDC records. In addition, taxa are only displayed if at least one public INSDC record is available.

- **Submittable Organism Names**
- **Carrying Out Taxon Checks**
- **Creating Taxon Requests**

### 17.1 Submittable Organism Names

Submitted organism names must be at ‘species’ rank. This rank type does not automatically mean the name is a published binomen (e.g. *Homo sapiens*): it is simply a rank, which differentiates the sequenced organism from another. For example, unidentified strains of the same bacterial genus should be kept as separate species, rather than binned together under the same genus name.

### 17.2 Carrying Out Taxon Checks

Interactive Webin features the ‘Taxonomy Check/Request’ tool, which allows submitters to check whether a taxon name is valid and submittable. You are advised to use this tool before beginning registration of any samples or data. If the name you enter is recognised, the tool will return a Tax ID: an integer unique to the matched name. A name may not be recognised if:

- You have provided a non-submittable taxonomic rank (e.g. genus, family)
- The name has not yet been classified by NCBI Taxonomy
If the name is not already available, you will need to request its addition, the process for which is described below. The taxonomists continually add both published species and informal names for novel species as sequences are deposited into the nucleotide archives.

Note that sometimes names have synonyms. This may happen if a name has been changed at some point and the old name is still recognised. More often, English common names for a species may be recognised. For example, if you input ‘human’, you will be offered ‘Homo sapiens’. If you enter ‘cat’, you are offered ‘Felis catus’. In any of these cases, the alternative is name is recognised and the most up-to-date scientific name is used. The majority of species do not have common names.

17.3 Creating Taxon Requests

If you wish to request a new taxon name, fill the details into the ‘Taxonomy Check/Request’ tool. If you have multiple names to request, use the ‘Add another organism’ button to do so: submissions of numerous requests instead of one multi-name request may be rejected at the discretion of helpdesk staff. Once your request is received, it will be reviewed and sent to the taxonomy service. You will be contacted again by ENA staff to let you know the names have been added, or else to advise you of corrections or changes to be made. Once this process is complete, the name will be indexed within our submission tool within 2 working days and you can then continue your submission.

Author citations should not be included in scientific names. Exceptions to this are only made where there are ambiguities between different nomenclature codes, e.g. *Agathis montana* Shest. 1932 and *Agathis montana* de Laub. 1969 refer to an insect and a plant, respectively.

Below are further rules specific to different categories of organism which should be adhered to when requesting new taxon names.

- **Unidentified/Novel Organisms**
  - Prokaryotes
  - Eukaryotes
- Environmental Samples
- Cyanobacteria
- Synthetic Sequences
- Viruses
- Endosymbionts

17.3.1 Unidentified/Novel Organisms

An informal name is used when the organism being submitted has not been identified to a specific name, or when a novel species name has not been published. The informal name can later be updated to a formal name, when the species is identified and appears in the literature. Creating an informal name is simple enough, and is often done in the format of <Submitter Initials>-<Year>, e.g. SH-2020. Below is further advice specific to prokaryotes/eukaryotes. Taxonomy is not simple and can be subject to many caveats and exclusions, and NCBI Taxonomy may make changes or provide feedback on a name.

**Prokaryotes**

If the genus is known but the species is novel or unidentified, please use the following format, where *identifier* is something unique to the culture, such as a strain/voucher ID, and is at least 3 characters long:
If the genus is new, please include its full lineage so it can be placed appropriately.

If the genus is unknown, please use the name of the highest known taxonomic rank, followed by the relevant descriptor (*bacterium* or *archaeon*), followed by the identifier:

<Rank> <bacterium or archaeon> <identifier>
Bacillaceae bacterium ABC123
Thermococcales archaeon DEF456

Note that if you are publishing a novel species, each strain of the proposed species should be given a unique informal name. Only after publication will the individual records be merged and renamed as the formal name.

Cyanobacteria should always be submitted with the strain appended, even when the species epithet is provided.

NCBI Taxonomy no longer adds prokaryotic binoma with the strain appended, in the form:

<Genus> <species> <strain>

If you wish to classify your submission this way, add the strain as a separate piece of sample metadata when submitting to us.

**Eukaryotes**

Higher organisms are treated similarly to prokaryotes, but the term ‘sp.’ is applied no matter which taxonomic rank is being used. If multiple strains/isolates/samples are identified to be from the same unidentified or novel species, they should be grouped as a single taxonomic name. For example, if three strains of *Candida* (ABC, DEF, and GHI) are identified as being from the same species, they should be given a single informal name to act as a placeholder. It is recommended that this be a number, followed by an author’s initials and the year.

<table>
<thead>
<tr>
<th>Informal Name</th>
<th>Informal Groupings</th>
</tr>
</thead>
<tbody>
<tr>
<td>Candida sp. ABC</td>
<td>Candida sp. 1 SH-2020</td>
</tr>
<tr>
<td>Candida sp. DEF</td>
<td>Candida sp. 2 SH-2020</td>
</tr>
<tr>
<td>Candida sp. GHI</td>
<td>Candida sp. 3 SH-2020</td>
</tr>
<tr>
<td>Candida sp. JKL</td>
<td></td>
</tr>
<tr>
<td>Candida sp. MNO</td>
<td></td>
</tr>
<tr>
<td>Candida sp. PQR</td>
<td></td>
</tr>
<tr>
<td>Candida sp. STU</td>
<td></td>
</tr>
<tr>
<td>Candida sp. VWX</td>
<td></td>
</tr>
</tbody>
</table>

If you are requesting a binomen which is already mentioned in published literature but has not been added to the database, then simply submitting this name with a paper reference will be sufficient.

### 17.3.2 Environmental Samples

If you are submitting a metagenomic sample (e.g. for metagenomic reads) there are numerous metagenomic taxa. You should use one of these for this purpose: visit the “metagenomes” tax node, click the *Tax tree* tab and use the arrow icons to expand the categories.
Use most appropriate environmental name for your metagenome. New names under this node are added only rarely and you should use the closest available choice. Only request a new name of this type if you are unable to use anything listed.

If you are submitting assembled/annotated sequences which are identified from homology alone with no prior culturing or isolation of the organism, this is considered an environmental sample. As an example, these may have been produced by 16S amplification of a metagenomic sample. Exceptions to this group include organisms which can be reliably recovered from their diseased host (e.g. endosymbionts, phyoplasmas) and organisms from samples which are readily identifiable by other means (e.g. cyanobacteria). Such organisms are not considered in the way described here.

Environmental samples are usually prefixed with the term uncultured and are not allowed to have a species epithet. Some examples of basic organism names that can be used include:

- uncultured bacterium (taxid:77133)
- uncultured archaeon (taxid:115547)
- uncultured cyanobacterium (taxid:1211)
- uncultured prokaryote (taxid:198431)
- uncultured fungus (taxid:175245)
- uncultured eukaryote (taxid:100272)

More granular identification is preferred, up to genus level. For prokaryotes the format is:

- uncultured <Rank> sp.
- uncultured Bacillus sp.
- uncultured Thermococcus sp.

For fungi, the ‘sp.’ is dropped:

- uncultured <Rank>
- uncultured Glomus
- uncultured Saccharomycetes
17.3.3 Cyanobacteria

Cyanobacteria taxonomy is very complex and so the strain or culture collection identifier is always captured as part of the organism name, whether or not it is identified at species level.

Nostoc punctiforme PCC 73102
Chroococcidiopsis sp. SAG 2025

17.3.4 Synthetic Sequences

Synthetic sequences, such as cloning and expression vectors, can use one of the relevant taxa:

- synthetic construct (taxid:32630) [uses translation table 11]
- eukaryotic synthetic construct (taxid:111789) [uses translation table 1]
- synthetic construct (code 6) [uses translation table 6]

Alternatively, a unique name can be requested. In such cases, a name is formed from the type of construct and a unique identifier. Some real examples of this:

- Cloning vector pNICO
- Expression vector pTEV5
- Site-specific excision vector pFLPe4

17.3.5 Viruses

Viruses do not fit well into biological classification systems and do not follow the format of binomial nomenclature. Instead, descriptive names are formed, usually referencing the host or disease.

NCBI Taxonomy will accept isolate names for well characterized isolates, isolates with commercial interests, or those with biological and medical implications. Certain viruses, specifically those involved in human health, should be named in accordance to known standards where metadata such as strain, host and serotype, are included in the taxon name. Some examples:

- HIV-1 CRF02_AG:08GQ032
- Norovirus 13-BH-1/2013/GII.17
- Norovirus 16-G0188/Ger/2016
- Norovirus groundwater/GII.17/61/2010/KOR
- Sapovirus Sewage/Toyama/Fu-Feb/2010/JP
- Sapovirus Hu/Toyama/Jan3519/2013/JP

However, note that this explicitly excludes Influenza: NCBI Taxonomy will no longer add new isolates for Influenza. A paper was published to explain this decision and is available at https://doi.org/10.7287/peerj.preprints.3428v1

17.3.6 Endosymbionts

Endosymbionts live within the cells of their host organisms and cannot usually be cultured outside the host. Although technically uncultured according to our terminology, they are exempt from the treatment of other environmental samples. Naming is usually in the format: “<type> endosymbiont of <host>”. Some real examples:

- endosymbiont of Acharax sp. [taxid:568145]
- bacterium endosymbiont of Donacia thallassina [taxid:742888]
- Wolbachia endosymbiont of Drosophila recens [taxid:214475]
- Rickettsia endosymbiont of Camponotus sayi [taxid:359403]
Metagenomic studies involve assembling sequencing data sampled from an entire biome all the way down to the individual species that were living in that environment. As a result, there are often many queries regarding how these assemblies are submitted in order to make the quality of the assembly and original source of data as clear as possible.

The following image illustrates the stages of a metagenome assembly study and what is submittable to each of the metagenome assembly levels in ENA:

Please see here for more information on how to submit your metagenome assemblies to ENA.
18.1 What is defined as a MAG within ENA?

Within ENA, a MAG is described as a single-taxon assembly based on one or more binned metagenomes asserted to be a close representation to an actual individual genome (that could match an already existing isolate or represent a novel isolate).

There should only be one MAG submitted for each species within a biome. This can be determined using a de-replication step or by choosing the highest quality representative genome for each predicted species.

MAG assemblies are registered within ENA in the same domain as cultured isolate genome assemblies which means that these genome assemblies are searchable alongside cultured isolates and are used by the same downstream processes. As an environmental sample can contain many duplicate genomes of the same organism and as MAG assemblies are more prone to contamination, we request only the highest quality unique-taxon submissions are submitted as MAGs. This means only the highest quality sequences and most accurate annotated features are used within these downstream services to keep them as relevant and informative as possible.

18.2 How is the quality of a metagenomic assembly defined?

When binned and MAG samples are registered with ENA, three quality measures are recorded which together determine the overall quality of a metagenome derived assembly. These three measures follow the data standards defined by the Genomic Standards Consortium (GSC) in the 2018 publication here.

The three attributes in the sample checklists that contribute to the overall quality are as follows:

1. Assembly Quality - a written description of the quality of an assembly (one of a choice of 3 options).
2. Completeness Score - the ratio of observed single-copy marker genes to total single-copy marker genes in chosen marker gene set (%).
3. Contamination Score - the ratio of observed single-copy marker genes in 2 copies to total single-copy marker genes in chosen marker gene set (%).

It is essential you complete these fields accurately so that the overall quality of an assembly is searchable within ENA. If you wish to search for binned metagenomes or MAGs by overall quality, the thresholds for these standards are outlined below.

**Finished Assembly**

Any assembly where the assembly quality is defined as: “Single contiguous sequence without gaps or ambiguities with a consensus error rate equivalent to Q50 or better”.

**High-quality draft**

An assembly with the following criteria:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>assembly quality</td>
<td>Multiple fragments where gaps span repetitive regions. Presence of the 23S, 16S and 5S rRNA genes and at least 18 tRNAs.</td>
</tr>
<tr>
<td>completeness score</td>
<td>&gt;90%</td>
</tr>
<tr>
<td>contamination score</td>
<td>&lt;5%</td>
</tr>
</tbody>
</table>

**Medium-quality draft**
An assembly with the following criteria:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>assembly quality</td>
<td>Many fragments with little to no review of assembly other than reporting of standard assembly statistics.</td>
</tr>
<tr>
<td>completeness score</td>
<td>50%</td>
</tr>
<tr>
<td>contamination score</td>
<td>&lt;10%</td>
</tr>
</tbody>
</table>

**Low-quality draft**

An assembly with the following criteria:

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>assembly quality</td>
<td>Many fragments with little to no review of assembly other than reporting of standard assembly statistics.</td>
</tr>
<tr>
<td>completeness score</td>
<td>&lt;50%</td>
</tr>
<tr>
<td>contamination score</td>
<td>&lt;10%</td>
</tr>
</tbody>
</table>

It is useful to bear in mind these quality thresholds when assembling and submitting metagenomic assemblies.

**18.3 How do I navigate through a metagenomics study?**

When submitting metagenomic samples it is important to enter the “sample derived from” field correctly. This field is used to correctly trace back your data through the assembly stages to its environmental biome-level origin. Users viewing the data can then also utilise the “sample derived from” attribute to navigate through the assemblies making your methods clear and reproducible.

It is important to note that you can access all the assembly layers from the study itself. However, these assemblies do not associate with each other. To look for associated metadata for these assemblies and which assemblies are derived from which, you should refer to the samples.

**18.4 How do I register samples for co-assemblies?**

When registering co-assemblies, you should reference multiple samples (or reads) in the “sample derived from” field. These can be formatted as one of the following:

A **comma separated list:**

This should not contain spaces.

e.g. formatted as

ERSxxxxxx,ERSxxxxxx

or
A range:
This should be in the case where the assembly was derived from many samples and should be formatted with a ‘-‘ character between two accession without any spaces. The accession format should be consistent within the range and all accessions referenced within the range should have been used in that assembly.

e.g. formatted as
ERSxxxxxx-ERSxxxxxx
or
ERRxxxxx-ERRxxxxx

If you wish to submit a primary assembly which is co-assembled from raw reads, please inform our helpdesk.

18.5 How do I submit uncultured virus genomes (UViGs)?

The method used for submission of uncultured virus genomes depends on the methods used to identify these genomes.

If the virus genome was derived from a study where the entire biome of environmental data was sequenced together and then binned by taxonomy, then please submit with the same methods as those outlined in the metagenome assembly submission guidelines. This is with the exception of the use of the GSC MIMAGS checklist. For virus genomes, the GSC MIUVIGS checklist should be utilised for each virus assembly.

If the virus genome was derived from a study using single-cell amplification techniques, then please submit with the same methods as those outlined in the environmental single-cell amplified genome assembly submission guidelines. This is with the exception of the use of the GSC MISAGS checklist. For virus genomes, the GSC MIUVIGS checklist should be utilised for each virus assembly.

18.6 How do I submit metagenome assemblies without raw data or primary assemblies to point to?

It is recommended to submit all levels of metagenomic assembly where possible. However, there are exceptions where this can not be done. For example, if you have assembled bacteria from a metagenome derived from a human host, your raw data may be contaminated with human DNA which you do not have the permission to make publicly available.

In cases where it is not possible to provide raw data or a primary metagenome, environmental samples should still be registered. However, as the registered environmental samples do not have any data associated with them, they need to be manually released to become available to the public.

If you have not submitted raw reads or primary assemblies, sample release can be done in advance of your study release without the risk of any data files being prematurely released. However, if you do not wish to have your sample metadata publicly available before your study is released, this option is not suitable and it is recommended to make a note of the Study release date and release these samples during the same time of the study release.

To manually release your environmental samples, first you need to prepare a submission XML file containing all your environmental sample accessions in a block of ACTION tags.

An example of a submission XML for the release of three environmental samples is below:

```
<SUBMISSION>
  <ACTIONS>
    <ACTION>
      (continues on next page)
    </ACTION>
  </ACTIONS>
</SUBMISSION>
```
These samples can then be released programmatically through the secure HTTPS protocol using a tool such as curl. Below is an example of a environmental sample release command:

```
curl -u username:password -F "SUBMISSION=@submission.xml" "https://www.ebi.ac.uk/ena/submit/drop-box/submit/
```

If your release is successful you should receive a receipt like the one below:

```
<RECEIPT receiptDate="2019-03-25T08:23:45.795Z" submissionFile="submission.xml" success="true">
  <MESSAGES>
    <INFO>sample accession "ERS3334823" is set to public status.</INFO>
    <INFO>sample accession "ERS3334824" is set to public status.</INFO>
    <INFO>sample accession "ERS3334825" is set to public status.</INFO>
    <INFO>Submission has been committed.</INFO>
  </MESSAGES>
  <ACTIONS>RELEASE</ACTIONS>
  <ACTIONS>RELEASE</ACTIONS>
  <ACTIONS>RELEASE</ACTIONS>
</RECEIPT>

18.6. How do I submit metagenome assemblies without raw data or primary assemblies to poin257 to?
19.1 What Are Locus Tags?

Locus tags are identifiers applied systematically to every gene in a sequencing project. If two submitters of different genomes use the same systematic names to describe different genes, this can be a source of confusion. Therefore, INSDC maintains a registry of locus tag prefixes to avoid overlap between genome annotation projects. The prefix is then used systematically to give a new unambiguous name to every gene.

19.2 How Do I Register A Locus Tag Prefix?

A locus tag prefix can be registered when a project is registered. Advice on doing this can be found at our study registration page. You can allow automatic assignment of a prefix, or you can specify your own.

If you did not add a prefix when you registered your project, it is possible to update the project with a prefix later on. See our advice on updating studies.

Please note that after you register the prefix, it will not be usable until 24 hours later. Therefore, you will not immediately be able to submit your annotated assembly.

A study can have multiple prefixes registered to it, but each prefix can only be registered to a single study.

19.2.1 What Rules Should The Prefix Conform To?

The prefix you register must follow the listed conventions or it will not be accepted:

- Must begin with a letter
- All letters must be upper case
- Must from 3 to 12 characters in length
- All characters must be alphanumeric with none such as -_*
Each prefix can only be registered to a single study; you cannot reuse a prefix from another study, or one which another user has registered.

### 19.3 How Do I Use Locus Tags?

Locus tags should be assigned to all genes, including both protein-coding and non-protein coding genes (e.g., structural RNAs). Within your flat file, you should include the `/locus_tag` qualifier in every gene, mRNA, CDS, 5'UTR, 3'UTR, intron, exon, tRNA, rRNA, misc_RNA, etc. We discourage the use of this qualifier on repeat_region and misc_features.

You should include the qualifier in each component of a gene, and it should have the same value for all components of a single gene. The feature itself should resemble the following in the format `<prefix>_<id>`:

```xml
<FT /locus_tag="BN5_00001"
```

Locus tags should be added systematically to every gene within a genome, generally in sequential order. When updating an assembly with additional annotation, you may either add new genes onto the end, or leave gaps when initially assigning /locus_tags and fill new genes into these gaps.

It is possible to encode information after the underscore but before the number. For example, the following tags might indicate the first genes of chromosomes I and II:

```
BN5_I00001
BN5_I100001
```

The following tags could indicate an rRNA gene and a tRNA gene:

```
BN5_r1112
BN5_t1113
```
Whenever possible, ENA provides access to two types of file for each run we present: the submitted file(s) and archive-generated file(s). Both are visible in the ENA Browser view for runs:

<table>
<thead>
<tr>
<th>Study Accession</th>
<th>Sample Accession</th>
<th>Experiment Accession</th>
<th>Run Accession</th>
<th>Scientific Name</th>
<th>FASTQ FTP</th>
<th>Submitted FTP</th>
</tr>
</thead>
<tbody>
<tr>
<td>PRJEB1973</td>
<td>SAMEA749883</td>
<td>ERX000007</td>
<td>ERR000001</td>
<td><em>Saccharomyces paradoxus</em> W7</td>
<td>ERR0000001_1.fastq.gz</td>
<td>062_s_3.fastq.gz</td>
</tr>
<tr>
<td>PRJEB1973</td>
<td>SAMEA749882</td>
<td>ERX000011</td>
<td>ERR000002</td>
<td><em>Saccharomyces paradoxus</em> Y96</td>
<td>ERR0000002_1.fastq.gz</td>
<td>045_s_5.fastq.gz</td>
</tr>
<tr>
<td>PRJEB1973</td>
<td>SAMEA749875</td>
<td>ERX000001</td>
<td>ERR000003</td>
<td><em>Saccharomyces paradoxus</em> Q31.4</td>
<td>ERR0000003_1.fastq.gz</td>
<td>030_s_4.fastq.gz</td>
</tr>
<tr>
<td>PRJEB1973</td>
<td>SAMEA749878</td>
<td>ERX000004</td>
<td>ERR000004</td>
<td><em>Saccharomyces cerevisiae</em> S280C</td>
<td>ERR0000004.fastq.gz</td>
<td>006_s_2.fastq.gz</td>
</tr>
<tr>
<td>PRJEB1973</td>
<td>SAMEA749877</td>
<td>ERX000003</td>
<td>ERR000005</td>
<td><em>Saccharomyces paradoxus</em> Q74.4</td>
<td>ERR0000005_1.fastq.gz</td>
<td>045_s_6.fastq.gz</td>
</tr>
</tbody>
</table>

This page serves to briefly discuss the reason for this and the differences between the submitted and archive-generated files.
20.1 Submitted Files

The submitted files for any given run are copies of the files originally provided to us by the submitter. These files always undergo validation appropriate to their format, and are presented as-submitted with no automated curation. Formats are varied, and may be FASTQ but could also be others including BAM, FAST5, HDF5, etc.

20.2 Archive-Generated Files

Providing archive-generated FASTQs for runs is a means of bringing some consistency to the data we provide. By imposing a level of uniformity on these files, we can ensure users know what to expect of them and may incorporate them into pipelines with minimal friction.

Note that archive-generated FASTQ will not be available in the following uncommon scenarios:

- BAM/CRAM files containing @PG:longranger
- BAM/CRAM files containing @PG:cellranger
- Complete genomics native (data folder) submissions
- PacBio native (HDF5) submissions
- Many ONT native format submissions

20.2.1 Generated FASTQ Files

The number of files generated and their content varies depending on the nature of the submitted files
<table>
<thead>
<tr>
<th>Number of Application Reads</th>
<th>FASTQ Files</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>(&lt;\text{run_accession}&gt;).fastq.gz &lt;\text{run_accession}&gt;_1.fastq.gz</td>
<td>For experiments with single application reads all reads will be made available in one fastq file.</td>
</tr>
<tr>
<td></td>
<td>(&lt;\text{run_accession}&gt;_1.fastq.gz &lt;\text{run_accession}&gt;).fastq.gz &lt;\text{run_accession}&gt;_2.fastq.gz</td>
<td>Paired experiments with two application reads will be made available in 1-3 FASTQ files. For a paired experiment submitted with both application reads the first reads will be in (&lt;\text{run accession}&gt;_1.fastq.gz \file, the second reads will be in (&lt;\text{run accession}&gt;_2.fastq.gz, and any unpaired reads will be in (&lt;\text{run accession}&gt;).fastq.gz file. If files from a paired experiment are submitted and all reads are unpaired then only a single file is created: (&lt;\text{run accession}&gt;).fastq.gz</td>
</tr>
<tr>
<td>&gt; 2</td>
<td>(&lt;\text{run_accession}&gt;_N.fastq.gz )</td>
<td>For experiments with more than two application reads (e.g. Complete Genomics) one fastq file is created for each application read, however, no empty fastq files are created.</td>
</tr>
<tr>
<td>N/A</td>
<td>(&lt;\text{run_accession}&gt;_\text{consensus.fastq.gz} )</td>
<td>ONT or PacBio consensus reads.</td>
</tr>
<tr>
<td>N/A</td>
<td>(&lt;\text{run_accession}&gt;_\text{subreads.fastq.gz} )</td>
<td>PacBio subreads.</td>
</tr>
</tbody>
</table>
FASTQ File Format

```
@<run accession>.<spot index> [<spot name>]/<read index>
<bases>
+<phred qualities, ASCII encoded starting with '!' (33)>
```

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;run accession&gt;</td>
<td>The run accession. A spot is identified uniquely by the combination of the run accession and the spot index.</td>
</tr>
<tr>
<td>&lt;spot index&gt;</td>
<td>A positive integer assigned to the spots in the order in which they appear in the run. A spot is identified uniquely by the combination of the Run accession and the spot index.</td>
</tr>
<tr>
<td>&lt;spot name&gt;</td>
<td>The spot name as it was provided by the submitter. In cases where the read name is missing or was removed by the archive this field is not present.</td>
</tr>
<tr>
<td>&lt;read index&gt;</td>
<td>A positive integer assigned to the application reads in the order in which they appear in the spot: /1 for first application read and /2 for the second application read. In cases where the read name is missing or was removed by the archive this field is not present.</td>
</tr>
</tbody>
</table>

**Examples**

Single layout:

```
@ERR000017.1 IL6_554:7:1:249:322
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
+
???????????????????????????????????????>>>>
```

Paired (first read):

```
```
@ERR005143.1 ID49_20708_20H04AAXX_R1:7:1:41:356/1
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
+
hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh

Paired (second read):
@ERR005143.1 ID49_20708_20H04AAXX_R1:7:1:41:356/2
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
+
hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh

Single layout without read names:
@ERR000017.1
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
+
?????????????????????????????????????????????????>>>>>>

Paired without read names (first read):
@ERR000017.1
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
+
hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh

Paired without read names (second read):
@ERR005143.1
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
+
hhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhhh
Members of the user community produce software to assist in the preparation and submission of files to ENA. Some go on to make these more widely available to other users.

The following are tools which are openly available. Please note that we are not responsible for the creation or maintenance of these tools and if you have queries about their use you should contact the creators directly. While we had no hand in their production and have not thoroughly reviewed them, we have heard from users of them that they have been valuable.

**EMBLmyGFF3** is a creation of staff at NBI Sweden. This tool can be used to convert an assembly in FASTA format along with associated annotation in GFF3 format into the EMBL flat file format which is the required format for submitting annotated assemblies to us. More information available in their paper: https://doi.org/10.1186/s13104-018-3686-x

**Artemis Comparison Tool (ACT)** is a part of the Artemis Software developed at the Sanger Institute. Among other things, it can output your file in EMBL flat file format.

**EMBOSS Seqret** is a tool developed by colleagues at EBI. It can convert between many different formats for various different molecules.
CHAPTER 22

Introductory Webinar