Collaborative OPen Omics (COPO) Project Documentation

Release 1.0

The Collaborative OPen Omics (COPO) project team

Jan 22, 2024

GETTING STARTED

1 Getting Started **3**
- 1.1 Accessing COPO Website . . . 3

2 Projects Brokered through COPO **7**
- 2.1 Primary Projects . . . 7
- 2.2 Secondary Projects . . . 7

3 New User **9**
- 3.1 First time login . . . 9

4 Sample Managers **11**
- 4.1 Accessing the Accept or Reject Samples web page . . . 11
- 4.2 Guidelines for Sample Managers Assigned to More than one Manifest Group . . . 12
- 4.3 Accepting or Rejecting Samples . . . 13
- 4.4 Download Submitted Permits . . . 19
- 4.5 View Submitted Images . . . 20
- 4.6 Upload Manifest or Update Submitted Manifest on behalf of Manifest Submitters . . . 22

5 Stand-alone Profile **27**
- 5.1 Steps to Create a Stand-alone Profile . . . 27
- 5.2 Stand-alone Profile Components . . . 30

6 Tree of Life Profile **33**
- 6.1 Steps to Create a Tree of Life Profile . . . 33
- 6.2 Tree of Life Profile Components . . . 41

7 Sharing Profiles **43**
- 7.1 Sharer . . . 43
- 7.2 Sharee . . . 46

8 Sorting Profiles **47**

9 Profile Types Legend **49**

10 General Tree of Life Visual Submission Documentation **51**

11 Assembly Submission **53**
- 11.1 How to Submit Assemblies . . . 53
- 11.2 How to Delete Assemblies . . . 56

12 Barcoding Manifest Submission **59**
- 12.1 How to Submit Barcoding Manifests . . . 59

12.2 How to Delete Barcoding Manifests . . . 64

13 Files Submission **67**
13.1 How to Submit Files . . . 67
13.2 How to Delete Files . . . 75

14 Images Submission **77**
14.1 How to Submit Images . . . 77
14.2 View Submitted Images . . . 78

15 Permits Submission **81**
15.1 How to Submit Permits . . . 81
15.2 Download Submitted Permits . . . 83

16 Reads Submission **87**
16.1 How to Submit Reads . . . 87

17 Samples Submission **95**
17.1 How to Download Submitted Sample Manifest . . . 95

18 Sequence Annotations Submission **97**
18.1 How to Submit Sequence Annotations . . . 97
18.2 How to Delete Sequence Annotations . . . 100

19 Updating Profiles **105**
19.1 Deleting Profiles . . . 107

20 Updating Samples **111**

21 Updating Sequence Annotations **115**

22 Updating Reads **119**

23 Adding Children Projects to Parent Projects **123**
23.1 In COPO . . . 123
23.2 In Public Repositories . . . 123

24 Downloading Manifest Templates & Standard Operating Procedures (SOPs) **125**
24.1 Downloading Manifest Templates . . . 125
24.2 Recording Data in Manifests . . . 126

25 Using COPO API **127**
25.1 API End-points . . . 127

26 Exploring Manifest Wizard **139**
26.1 Prefill Manifests . . . 140

27 Tree of Life Dashboard **145**
27.1 Tree of Life Inspection . . . 147
27.2 Tree of Life Inspection by Genome Acquisition Lab . . . 147

28 Retrieving Accessions **149**

29 COPO Statistics **151**
29.1 Statistics over Time . . . 151
29.2 Statistics by Variable . . . 152

30 Manifest Checklist **155**
30.1 Barcoding Manifest Checklists . . . 155
30.2 Sample Manifest Checklists . . . 155

31 Setting Up COPO Project Locally with Docker **159**
31.1 Install Docker . . . 159
31.2 Initialise Docker Swarm . . . 159

32 Setting Up COPO Project Locally without Docker **175**
32.1 Set Up COPO Project on Linux . . . 175
32.2 Set Up COPO Project on Windows . . . 180

33 Brokering Data in COPO with FAIR Data Principles **183**
33.1 Findable . . . 183
33.2 Accessible . . . 183
33.3 Interoperable . . . 184
33.4 Reusable . . . 184
33.5 FAIR Webinars . . . 184

34 Licence **185**

35 Contributor Covenant Code of Conduct **187**
35.1 Our Pledge . . . 187
35.2 Our Standards . . . 187
35.3 Enforcement Responsibilities . . . 188
35.4 Scope . . . 188
35.5 Enforcement . . . 188
35.6 Enforcement Guidelines . . . 188
35.7 Attribution . . . 189

36 Privacy Notice **191**
36.1 Consent . . . 191
36.2 Data Access . . . 192
36.3 Data Retention . . . 192
36.4 Cookies . . . 192
36.5 Questions . . . 193

37 Terms of Use **195**

38 Contact COPO **197**

39 Frequently Asked Questions **199**
39.1 Dashboard . . . 199
39.2 Manifest Update . . . 200
39.3 Permits . . . 200
39.4 Profiles . . . 200
39.5 Samples . . . 204
39.6 Sample Managers . . . 205

40 Glossary **207**

Index **213**

Welcome to the guidelines for submission, updates and retrieval for the Collaborative OPen Omics project. Please use the links to find instructions specific to your needs.

CHAPTER

ONE

GETTING STARTED

Collaborative OPen Omics (COPO) is a website portal where scientists describe their research objects (raw or processed data, reads, samples, images etc.) using community-sanctioned metadata sets and vocabularies[1].

As a metadata broker, COPO encourages data generators to adhere to FAIR (Findable, Accessible, Interoperable and Reusable)[2] principles when submitting research objects, using semantic terms to add meaning to them and specify relationships between them.

COPO uses public or institutional repositories like European Nucleotide Archive (ENA) and National Center for Biotechnology Information (NCBI) to share the research objects to the wider scientific community.

COPO is developed by a team of Research Software Engineers at the Earlham Institute. Contributions are welcomed in the GitHub repository. All contributors must abide by the code of conduct described in the *Contributor Covenant Code of Conduct* section.

Visit the COPO website to make manifest submissions. An ORCID ID is required to access the application.

The website requires cookies, fonts and the limited processing of your personal data in order to function well. By using the portal you are consenting to this. To find out more about how COPO uses cookies and your personal data, please see our *Privacy Notice* and *Terms of Use*.

Hint: You can register for an ORCID ID on the ORCID website for free.

1.1 Accessing COPO Website

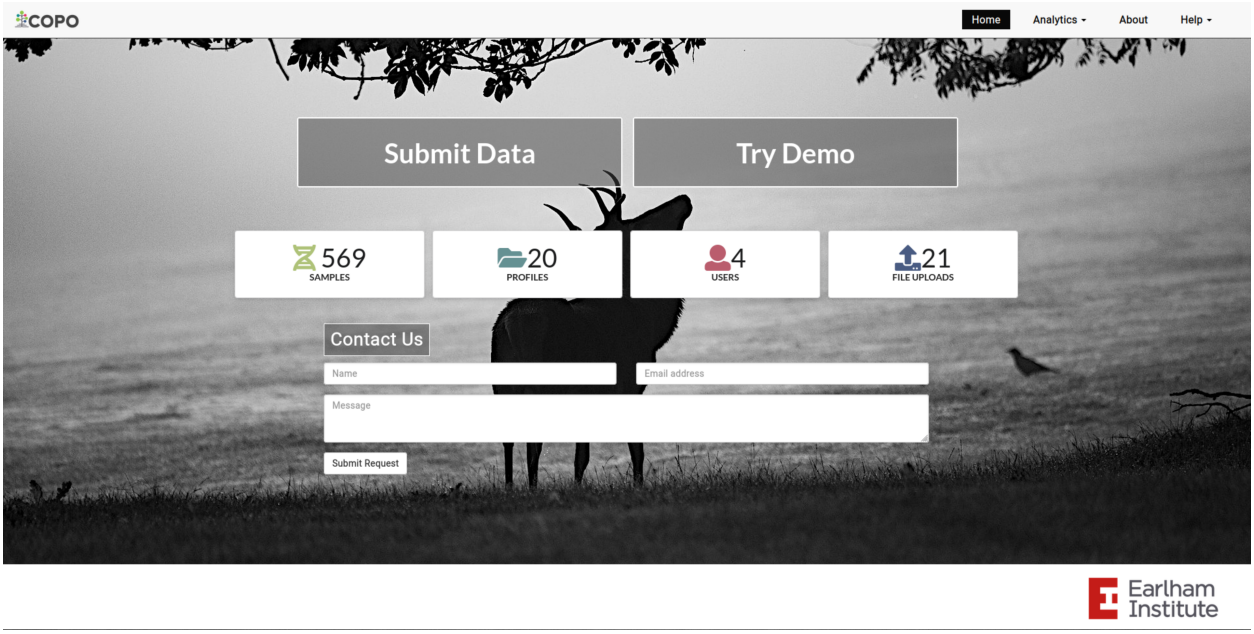
There are two ways to access the COPO website: **Demo** mode and **Production** mode.

Navigate to COPO live website to access the production mode **or** navigate to the COPO demo website to access the demonstration mode.

Note:

- **Production mode** is the live version of the COPO application. It is recommended to submit real data to the production mode because the submissions are made to the final repository.
- **Demo mode** is a testing environment for the COPO application. It is not recommended to submit real data to the demo mode because any submissions done are considered test submissions and there will be no submissions to the final repository.

[1] SOPs (Standard Operating Procedures) guide the metadata sets and vocabularies for manifests. See: *SOP guidelines*.
[2] See: *COPO FAIR data principles*.



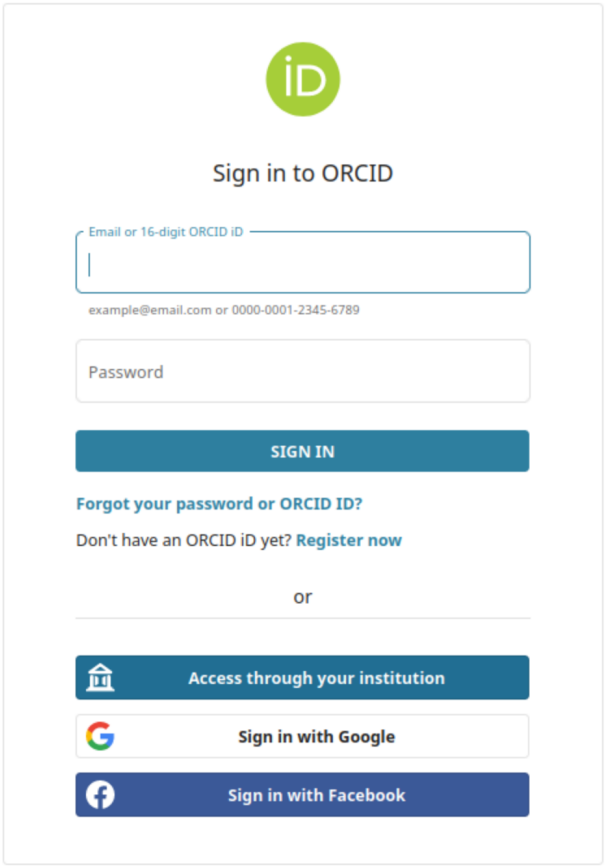
COPO homepage: Click `Submit Data` button to proceed to the ORCID sign-in form (as shown below)

Alternatively, click `Try Demo` button to access the demo mode.

COPO homepage: Click `Sign in with Orcid.org` button to proceed to the ORCID sign-in form (as shown below)

Orcid sign-in form: Enter your Orcid login credentials or click `Register now` to sign-up for an ORCID account.

If the login is successful, a redirection is made to the COPO application.



ORCID
The text of this website is published under a CC0 license. Images and marks are subject to copyright and trademark protection.
About ORCID Privacy Policy Terms of Use Accessibility Statement
ORCID Help Center Dispute procedures Brand Guidelines Cookie Settings

CHAPTER

TWO

PROJECTS BROKERED THROUGH COPO

Hint:

- Primary projects are also known as umbrella projects or main projects and are the main focus of the project. In COPO, primary projects are regarded as **Profile type** or ToL (Tree of Life)[1] project.
- Secondary projects are also known as subprojects or child projects and are part of the main project. In COPO, secondary projects are regarded as **Associated profile type** or associated ToL project.

2.1 Primary Projects

- Aquatic Symbiosis Genomics (ASG)[2]
- Darwin Tree of Life Samples (DToL)[3]
- Darwin Tree of Life Environmental Samples (DToL_ENV)
- European Reference Genome Atlas (ERGA)[4]
- Stand-alone

2.2 Secondary Projects

Note: In addition to the secondary projects outlined below, all projects listed under **Primary Projects** except Stand-alone, can be added as a subproject of a larger project.

- Biodiversity Genomics Europe (BGE)
- European Reference Genome Atlas - Pilot (ERGA_PILOT)
- European Reference Genome Atlas - Satellites (ERGA_SATELLITES)
- Population Genomics (POP_GENOMICS)

See also:

- *Standard Operating Procedure (SOP)* for the various projects
- *Manifest template* for the various projects
- *Using manifest wizard to prefill manifests*

[1] See term: *Tree of Life (ToL)*
[2] See term: *ASG*.
[3] See term: *DToL*.
[4] See term: *ERGA*.

CHAPTER

THREE

NEW USER

3.1 First time login

If signing in for the first time, a prompt requesting for an email address is displayed.

Enter an email address to proceed. A valid email address and agreement lead to the main web page that comprises all the **work profiles** associated with a user.

The email address supplied will be associated with the user's COPO account and will be used in COPO for tasks such as depositing data to remote repositories involving the user.

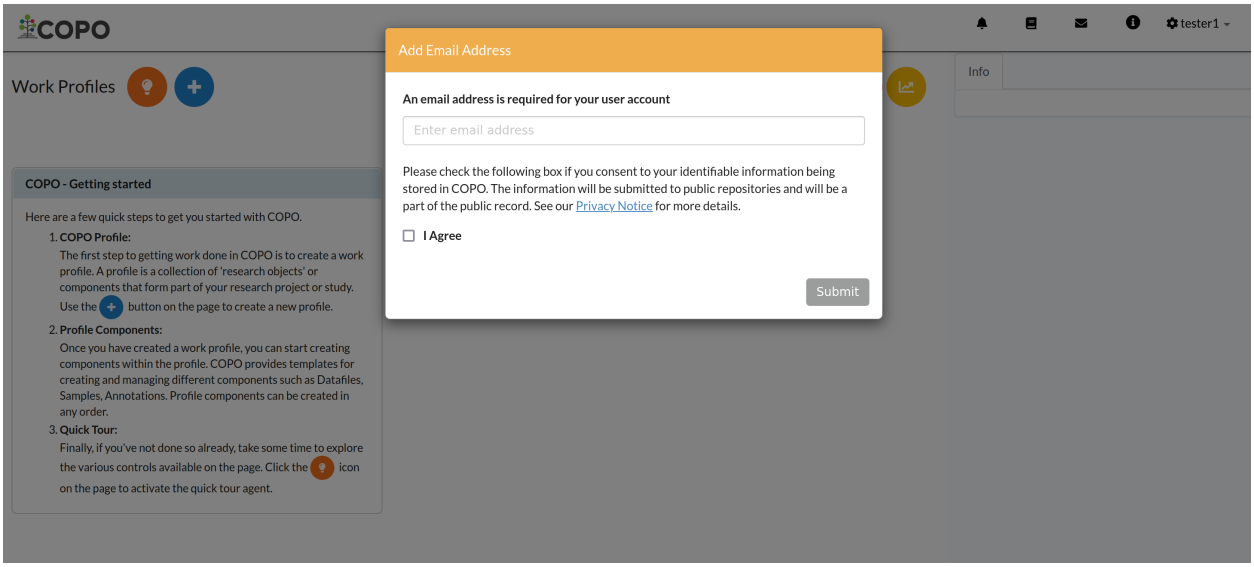


Fig. 1: **First time user: Email address prompt shown when a user logs into COPO for the first time**

CHAPTER

FOUR

SAMPLE MANAGERS

This section gives an overview of the COPO features that are delegated to Biodiversity Genomics Europe (BGE) checker or sample managers[1] who are responsible for accepting or rejecting samples produced by sample submitters[2].

Note:

- Make a request to the indicating the type of profile manifest group that you would like to be assigned to as a sample manager BGE (Biodiversity Genomics Europe).

  The permission will be granted after the request has been approved.

See also:

- *View images produced by sample submitters*
- *Download permits produced by sample submitters*
- *Sample Managers' Frequently Asked Questions*

4.1 Accessing the Accept or Reject Samples web page

Note: The button will only appear on the web page if you are granted permission to be a sample manager.

Once the button is clicked, the **accept/reject samples** web page will be displayed.

If you have been granted permission as a sample manager, you will receive an email notifying you that sample(s) has been submitted after a user uploads a manifest.

Similarly, you will receive an email when samples have been rejected.

Click the button to accept or reject samples.

Alternatively, navigate to Accept/Reject Samples' web page.

[1] See term: *Sample manager*

[2] See term: *Sample submitter*. Sample submitter may also be referred to as a manifest provider or manifest submitter.

4.2 Guidelines for Sample Managers Assigned to More than one Manifest Group

Note: The manifest dropdown menu will only be displayed on the **Accept or Reject samples** web page if you as a sample manager, belongs to more than one sample manager manifest group.

Hint: If the **dtol** sample manager group dropdown menu option is selected, both Aquatic Symbiosis Genomics (ASG)[3] profiles and Darwin Tree of Life (DToL)[4] profiles will be displayed in the **All profiles** tab (if any exists).

If you have been granted permission to be a sample manager for more than one manifest group, you can accept or reject samples for more than one manifest group by following the steps below:

1. Click the dropdown menu displayed beside the **Choose to Accept or Reject** web page title on the left side of the **Accept or Reject Samples** web page as shown below:

Fig. 1: **Accept or Reject Samples web page: Manifest dropdown menu beside 'Choose to Accept or Reject' web page title**

2. A list of all the manifest groups that you have been assigned to is then displayed.

   Choose the desired manifest group from the dropdown menu as shown below:

3. To **accept** samples, see the *Accepting samples* section

   OR

   To **reject** samples, see the *Rejecting samples* section

[3] See term: *ASG*.
[4] See term: *DToL*. *DToL* may sometimes be referred to as *DTOL*.

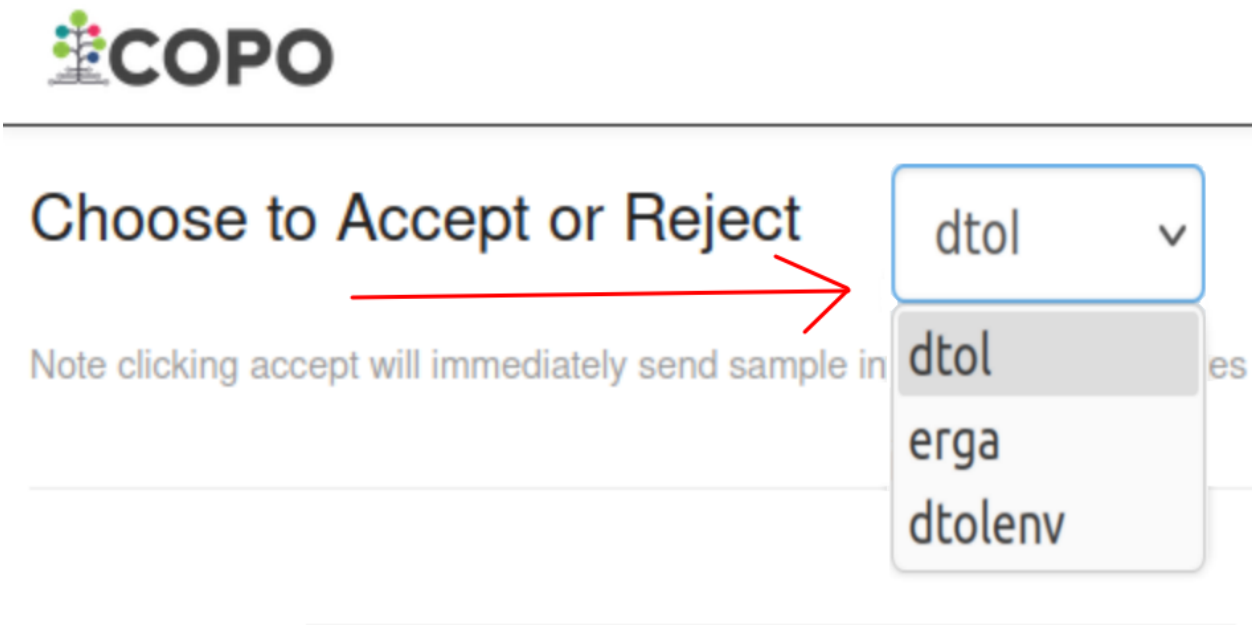


Fig. 2: **Accept or Reject Samples web page: Manifest dropdown menu with options displayed**

4.3 Accepting or Rejecting Samples

Note: A **Samples** table record will only be displayed if the desired profile that is clicked/highlighted on the left of the web page has submitted samples.

Hint:

- See *Accept or Reject Samples for more than one manifest group* section for guidance if you are assigned to more than one manifest group and would like to accept or reject samples.

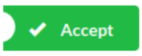
4.3.1 Accepting samples

The following sections describe how to accept samples depending on the type of manifest group that you have been assigned as a sample manager to:

- *Accepting ASG or DTOL samples*
- *Accepting ERGA samples as a BGE checker*
- *Accepting ERGA samples as a Sample Manager*

4.3.2 Accepting ASG or DTOL samples

1. Choose desired profile on the left of the **Accept or Reject Samples** web page
   - From the **All Profiles** tab if desired profile is an ASG (Aquatic Symbiosis Genomics)[Page 12, 3] or DTOL (Darwin Tree of Life)[Page 12, 4] profile
2. In the **Pending Samples** tab, select desired sample record(s) by clicking the checkbox(es) associated with the sample record(s).

   Then, click the 

button as shown below.

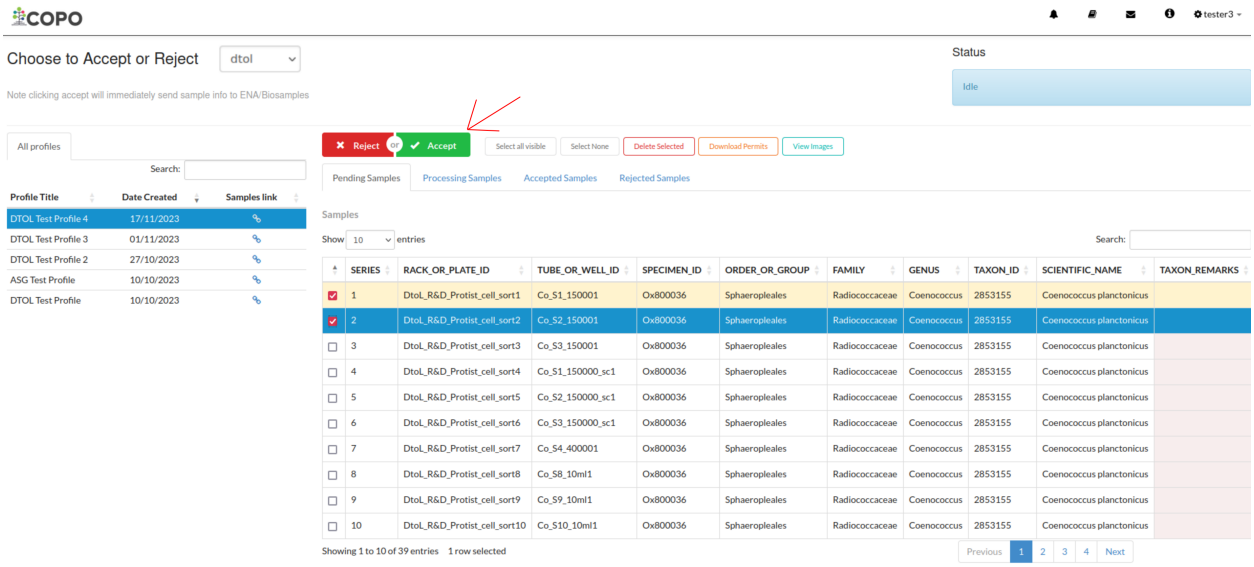


Fig. 3: **Accept or Reject Samples web page: Accepting samples within ASG or DTOL profiles**

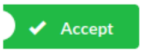
4.3.3 Accepting ERGA samples as a BGE checker

Note:

- In the **Samples** table, sample record(s) will only be displayed in the **Pending BGE Checker Samples** tab, if you are assigned as a **Biodiversity Genomics Europe (BGE) checker** for the desired profile.
- All COPO ERGA (European Reference Genome Atlas)[5] profiles are displayed in the **All Profiles** tab but submitted samples within them cannot be accepted or rejected.

---

1. Choose desired profile on the left of the **Accept or Reject Samples** web page
   - From the **Profiles for My Sequencing Centre** tab
2. In the **Pending BGE Checker Samples** tab, select desired sample record(s) by clicking the checkbox(es) associated with the sample record(s).

   Then, click the 

button as shown below.

[5] See term: *ERGA*.

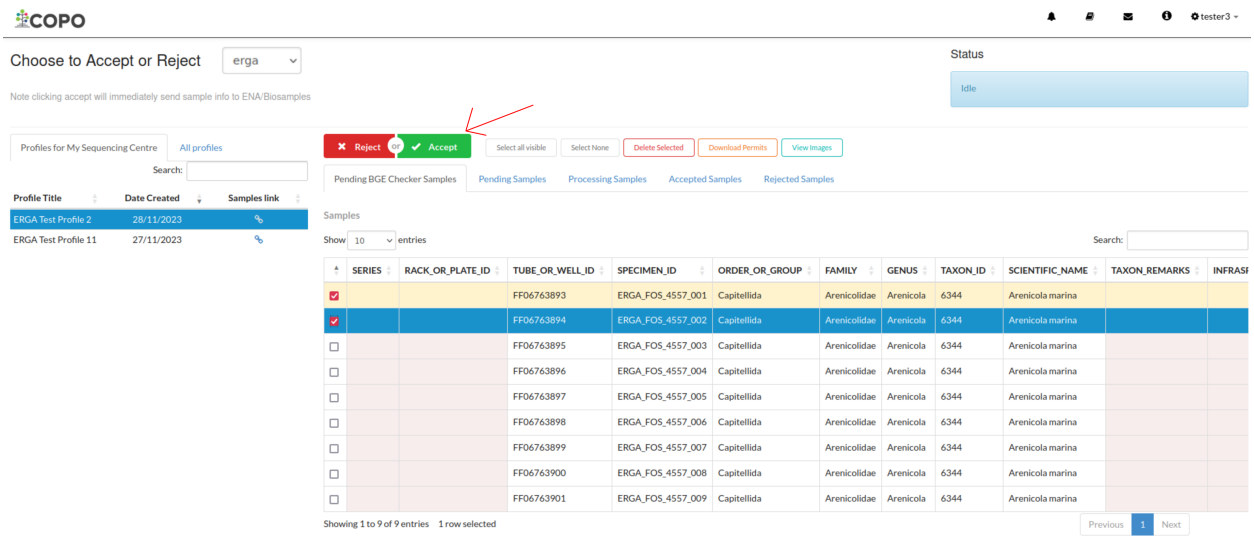
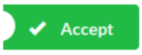


Fig. 4: **Accept or Reject Samples web page: Accepting samples within ERGA profiles as a Biodiversity Genomics Europe (BGE) checker**

3. In the **Pending BGE Checker Samples** tab, select desired sample record(s) by clicking the checkbox(es) associated with the sample record(s).

   Then, click the 

button as shown below.

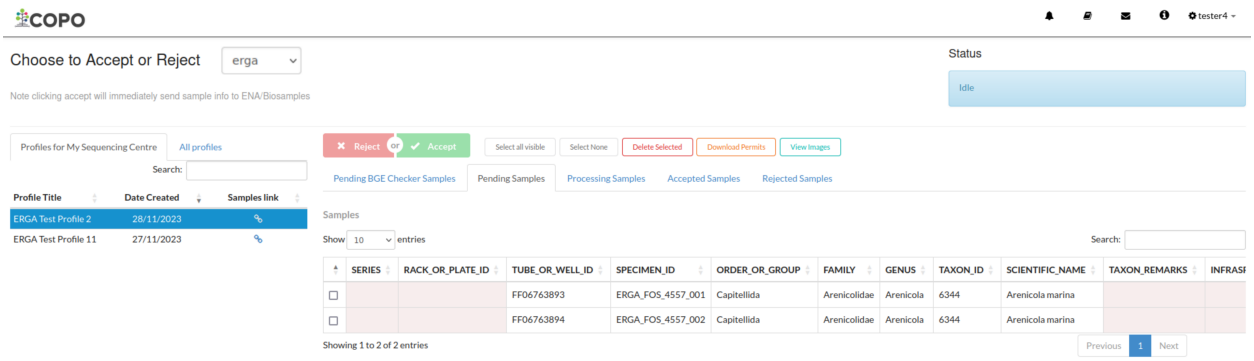


Fig. 5: **Accept or Reject Samples web page: Samples accepted by Biodiversity Genomics Europe (BGE) checker are displayed in 'Pending Samples' tab**

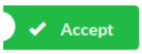
4.3.4 Accepting ERGA samples as a Sample Manager

Note:

- In the **Samples** table, sample record(s) will only be displayed in the **Pending Samples** tab for the desired profile **after** they have been accepted by a Biodiversity Genomics Europe (BGE) checker.
- All COPO ERGA[5] profiles are displayed in the **All Profiles** tab but submitted samples within them cannot be accepted or rejected.

Hint: Rejected samples are displayed in the **Rejected Samples** tab.

1. Choose desired profile on the left of the **Accept or Reject Samples** web page
   - From the **Profiles for My Sequencing Centre** tab
2. In the **Pending Samples** tab, select desired sample record(s) by clicking the checkbox(es) associated with the sample record(s).

   Then, click the 

button as shown below.

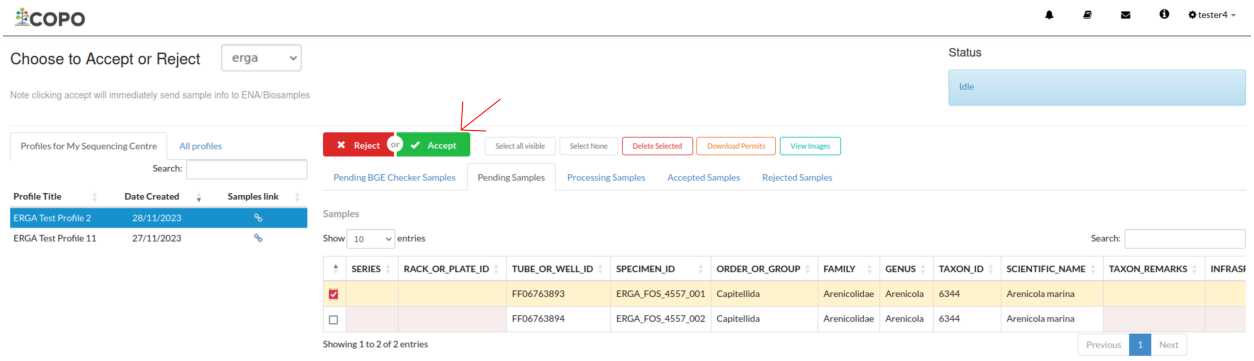


Fig. 6: **Accept or Reject Samples web page: Accepting samples within ERGA profiles**

A confirmation popup dialogue will be displayed as shown below.

Click the **Ok** button to accept the selected sample record(s) or click the **Cancel** button to cancel the action.

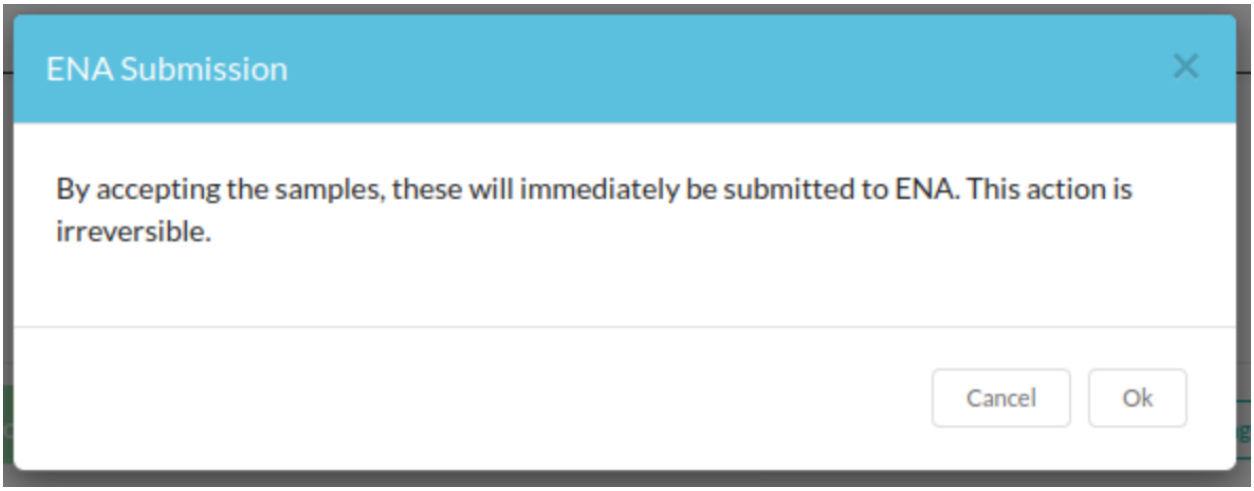


Fig. 7: **Accept or Reject Samples web page: Accept samples confirmation dialogue**

3. If the **Ok** button is clicked, the accepted samples will proceed to the processing stage and will be displayed in the **Processing Samples** tab as shown below:

OR

4. After the samples have been processed, the samples will proceed to the accepted stage and will be displayed in the **Accepted Samples** tab as shown below:

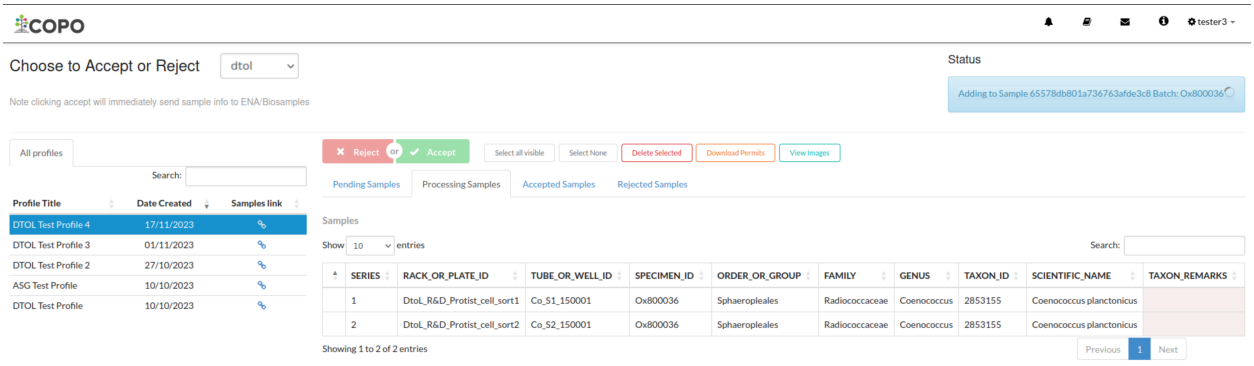


Fig. 8: **Accept or Reject Samples web page: Accepted ASG or DTOL samples at the processing stage**

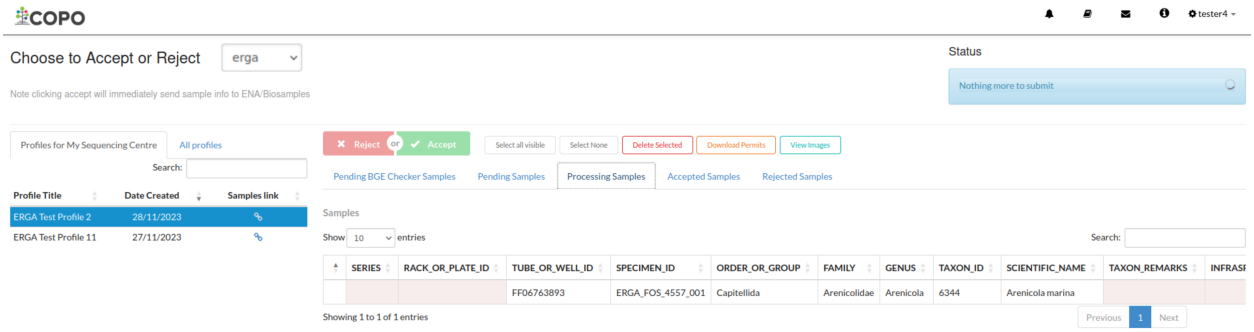


Fig. 9: **Accept or Reject Samples web page: Accepted ERGA samples at the processing stage**

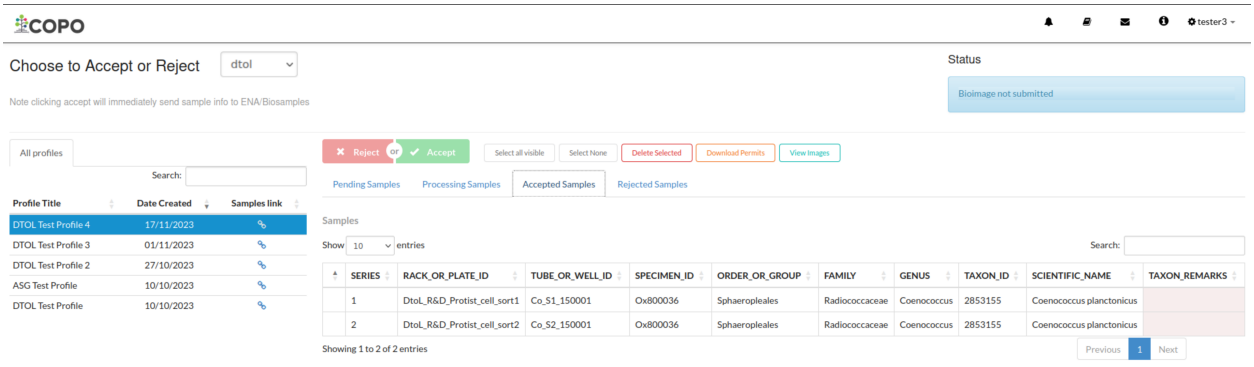


Fig. 10: **Accept or Reject Samples web page: Accepted ASG or DTOL samples at the accepted stage**

OR

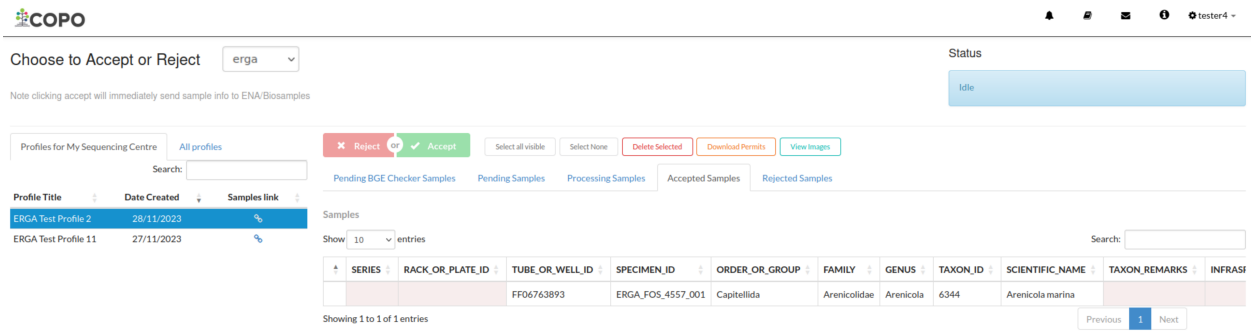


Fig. 11: **Accept or Reject Samples web page: Accepted ERGA samples at the accepted stage**

4.3.5 Rejecting samples

1. Choose desired profile on the left of the **Accept or Reject Samples** web page
   - From the **Profiles for My Sequencing Centre** tab - if desired profile is an ERGA[Page 14, 5] profile

   OR

   - From the **All Profiles** tab if desired profile is not an ERGA[Page 14, 5] profile
2. In the **Pending Samples** tab, select desired sample record(s) by clicking the checkbox(es) associated with the sample record(s).

   Similarly, if you are a Biodiversity Genomics Europe (BGE) checker, select desired sample record(s) from the **Pending BGE Checker Samples** tab.

   Then, click the

button as shown below:

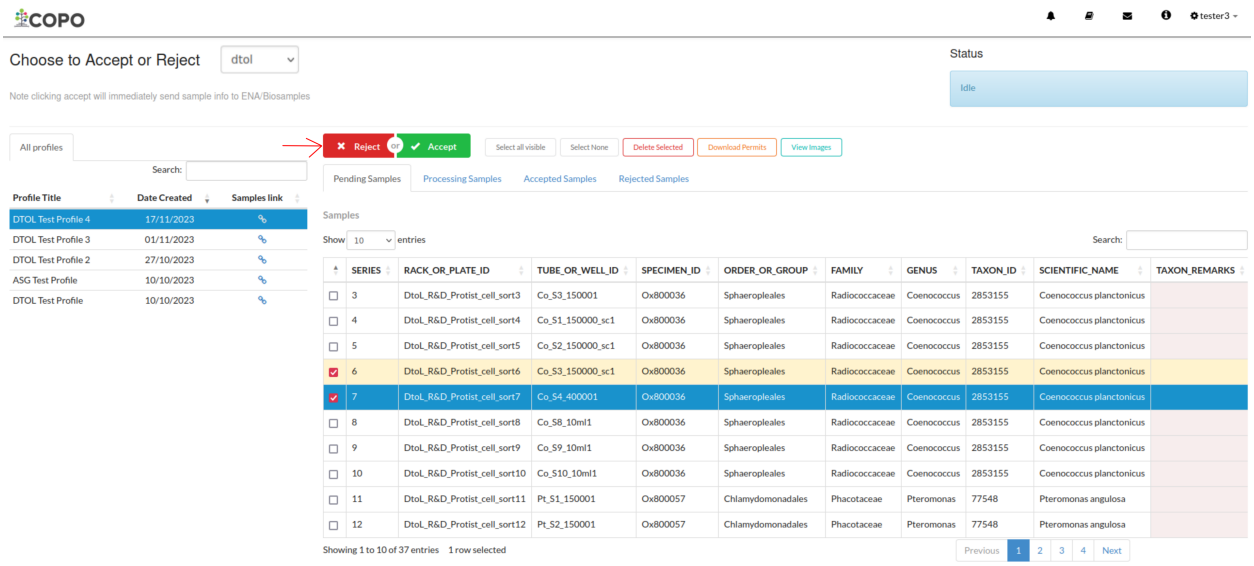


Fig. 12: **Accept or Reject Samples web page: Rejecting samples**

3. The samples will proceed to the rejected stage and will be displayed in the **Rejected Samples** tab as shown below:

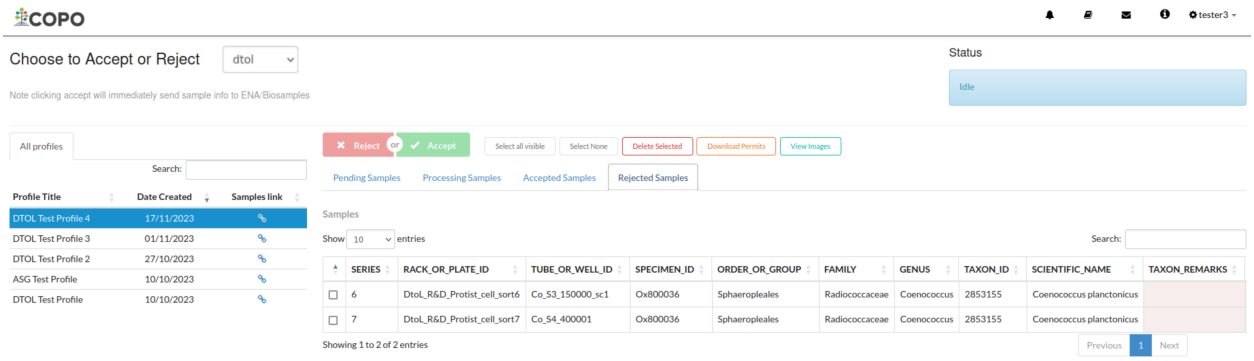


Fig. 13: **Accept or Reject Samples web page: Rejected samples at the rejected stage**

4.4 Download Submitted Permits

If you have been assigned as a **sample manager**, you can view the permits submitted for submitted sample(s) on the Accept/Reject Samples' web page by following the steps below:

Note:

- Permits can only be downloaded for submitted samples that are **pending** action by a sample manager or have been **accepted** by a sample manager.
- Permits exist for ERGA[Page 14, 5] profiles only.

Hint: To download submitted permits for samples within the **Accepted Samples** tab, `CTRL + Click` the desired sample record(s) then, click the Download Permits button to download permit(s) submitted for the selected record(s).

---

1. Navigate to the **Accept or Reject Samples** web page.

   See *How to access Accept or Reject Samples web page* section for guidance on how to access the **Accept or Reject Samples** web page.

2. Select the sample record(s) that you would like to download the permits for.

   Then, click the Download Permits button to download permit(s) submitted for the selected sample record(s).

3. If any permit submission(s) exist for the selected sample record(s), the permits will be automatically downloaded for the selected sample record(s) as shown below:

   Hint: Permits are downloaded as a `.zip` file

   If no permits were submitted for the selected sample record(s), a message is displayed in the popup dialogue indicating such as shown below:

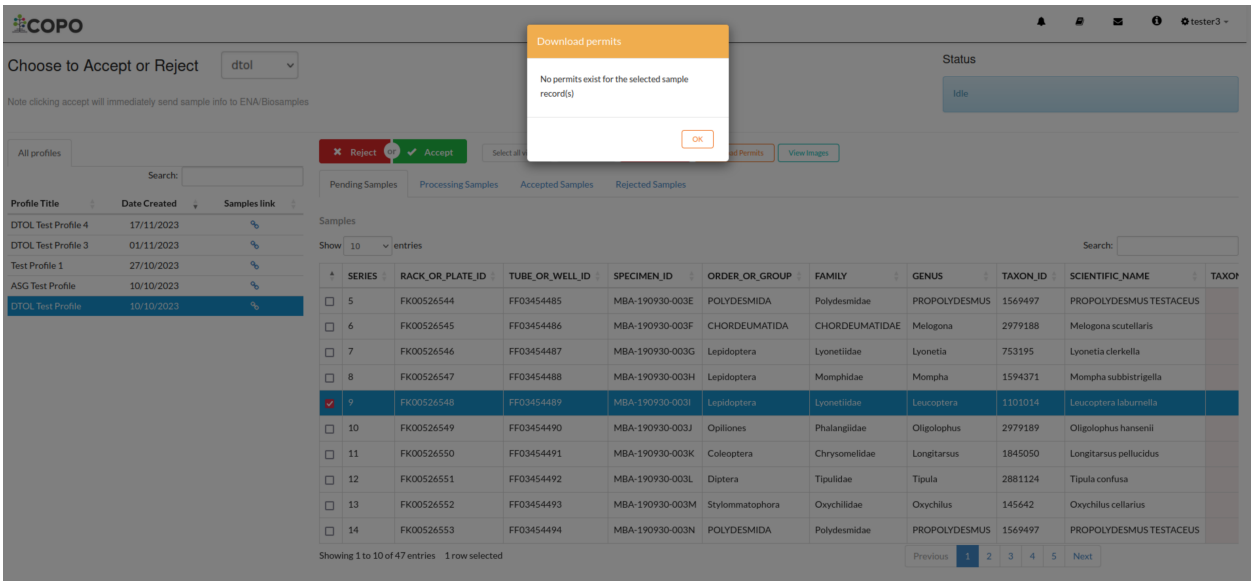


Fig. 14: **Accept or Reject Samples web page: Popup dialogue displaying message, 'No permits exist for selected sample record(s)'**

4.5 View Submitted Images

If you have been assigned as a **sample manager**, you can view the images submitted for submitted sample(s) on the Accept/Reject Samples' web page by following the steps below:

Hint: To view submitted images for samples within the **Accepted Samples** tab, `CTRL + Click` the desired sample record(s) then, click the View Images button to view image(s) submitted for the selected record(s).

---

1. Navigate to the **Accept or Reject Samples** web page.

   See *How to access Accept or Reject Samples web page* section for guidance on how to access the **Accept or Reject Samples** web page.

2. Select the sample record(s) that you would like to view the images for.

   Then, click the View Images button to view image(s) submitted for the selected sample record(s).

3. If any image submission(s) exist for the selected sample record(s), a popup dialogue will be displayed with the image(s) submitted for the selected sample record(s) as shown below:

   Hint: Click the image to view a larger version.

   OR

   If no images were submitted for the selected sample record(s), a message is displayed in the popup dialogue indicating such as shown below:

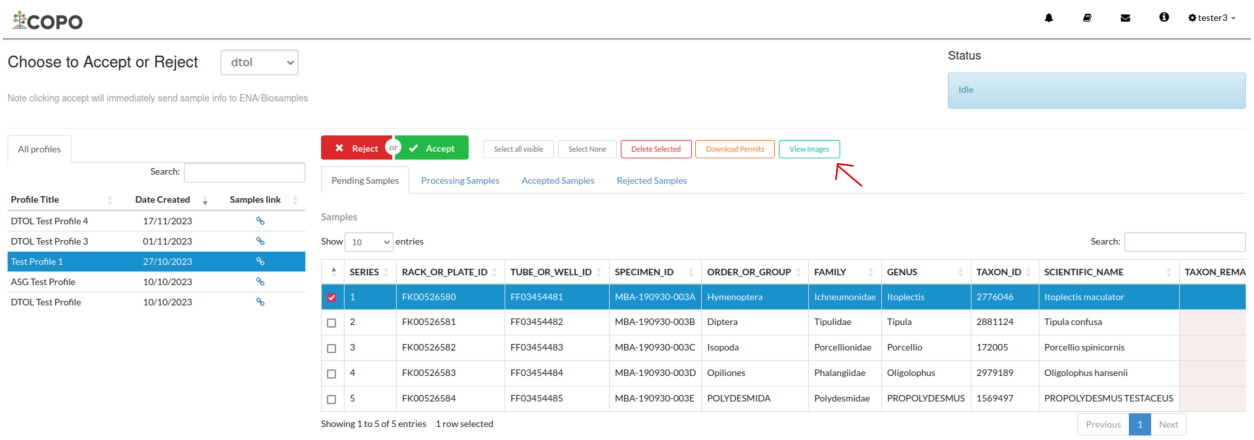


Fig. 15: **Accept or Reject Samples web page: Pointer to 'View images' button**

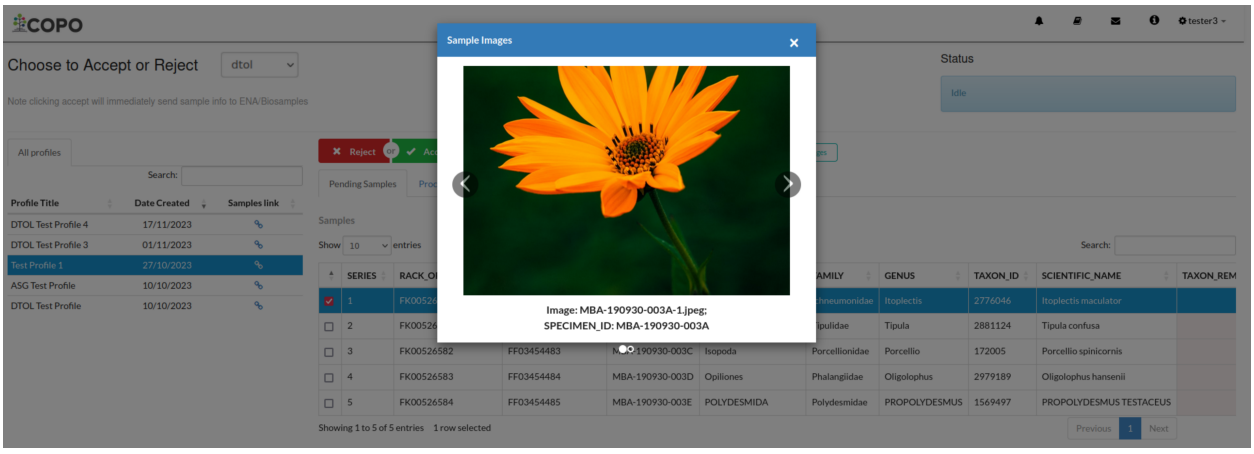


Fig. 16: **Accept or Reject Samples web page: Popup dialogue displaying submitted image(s) for selected sample record(s)**

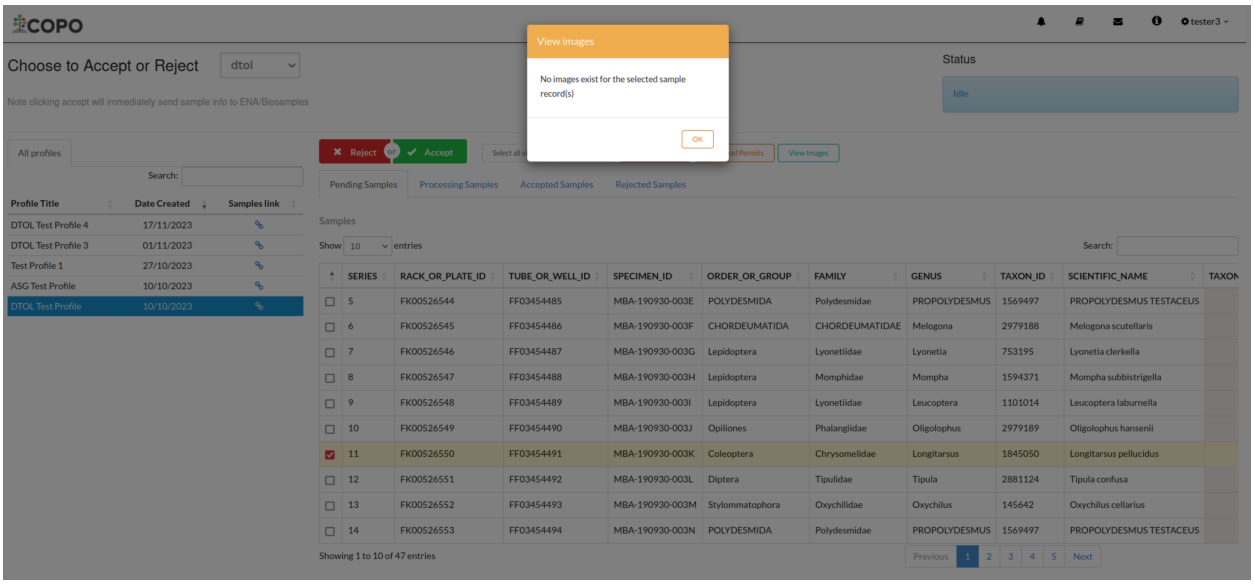


Fig. 17: **Accept or Reject Samples web page: Popup dialogue displaying message, 'No images exist for selected sample record(s)'**

4.6 Upload Manifest or Update Submitted Manifest on behalf of Manifest Submitters

---

Note:

- The manifest dropdown menu will only be displayed on the **Accept or Reject samples** web page if you as a sample manager, belongs to more than one sample manager manifest group.
- If the **dtol** sample manager group dropdown menu option is selected, both Aquatic Symbiosis Genomics (ASG) profiles and Darwin Tree of Life (DToL) profiles will be displayed in the **All profiles** tab (if any exists).
- The **Samples** table will only be displayed if the selected/highlighted profile has submitted samples.

Hint:

- See the *Updating Samples* section for information about which field values can be updated. Samples can be updated by resubmitting the manifest with the updated metadata.

The following steps can be followed to upload a manifest or update a submitted manifest on behalf of a manifest submitter[Page 11, 2]:

1. Navigate to the **Accept or Reject Samples** web page.

   See *How to access Accept or Reject Samples web page* section for guidance.

2. On the left of the **Accept or Reject Samples web page**, click the icon in the **Samples link** table column of the table row for the profile that you would like upload a manifest for or update a submitted manifest for as shown below:

   If the desired profile is an ASG profile or DTOL profile, navigate to the **Samples** web page by clicking the icon in the **Samples link** table column of the table row for the desired profile in the **All Profiles** tab as shown

below:

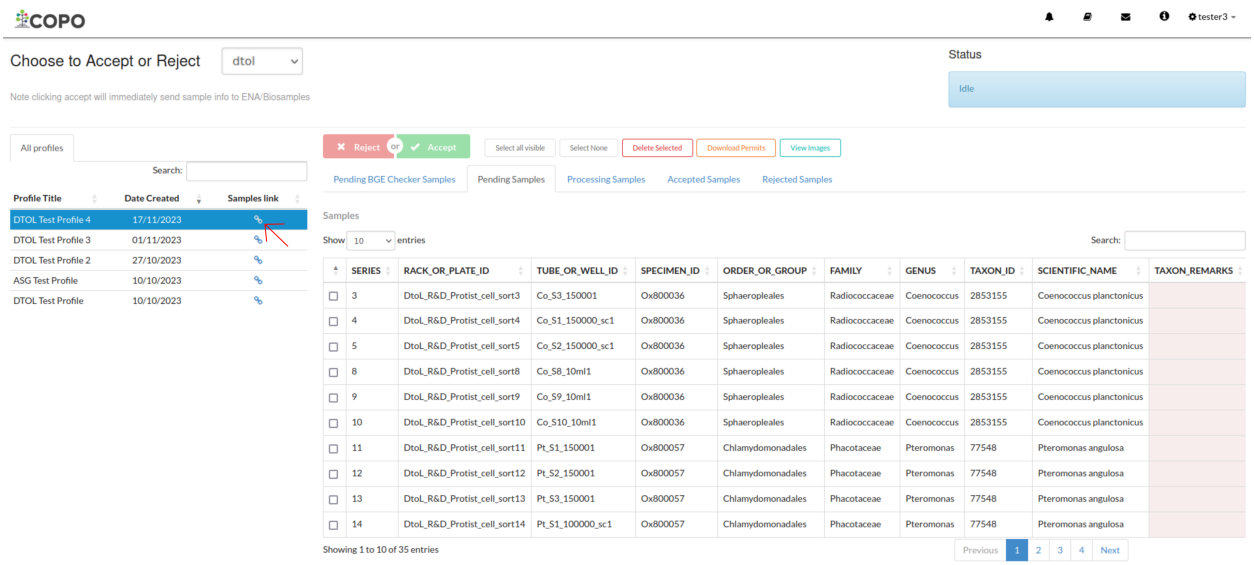


Fig. 18: **Accept or Reject Samples web page: Navigate to 'Samples' web page by clicking the link associated with a DTOL profile**

Similarly, if the desired profile is an ASG profile, the same action occurs.

OR

If the desired profile is an ERGA profile, navigate to the **Samples** web page by clicking the icon in the **Samples link** table column of the table row for the desired profile in the **Profiles for My Sequencing Centre** tab as shown below:

3. The **Samples** web page will be displayed as shown below:

   Similarly, if the desired profile is an ASG profile, the **Samples** web page will be displayed.

   OR

4. If you do not have the submitted manifest for the profile, see the *How to Download Submitted Sample Manifest* section for guidance on how to download the submitted sample manifest. Then, refer to the *Updating Samples* section for guidance on how to submit the modified manifest.

   OR

   If you have the submitted manifest for the profile, see the *Updating Samples* section for guidance on how to update the submitted manifest.

   OR

   If you would like to upload a newer version of a manifest that has already been submitted on behalf of the manifest provider, please follow the steps below:

   1. Download the submitted manifest by following the guidelines described in the *How to Download Submitted Sample Manifest* section if you do not have the newer version of the manifest with the sample metadata information.
   2. Transfer the sample metadata from the submitted manifest into the newer version of the manifest.
   3. Send an email to the COPO team at , indicating the profile type as well as the profile title and requesting that the samples be **removed** from the profile.

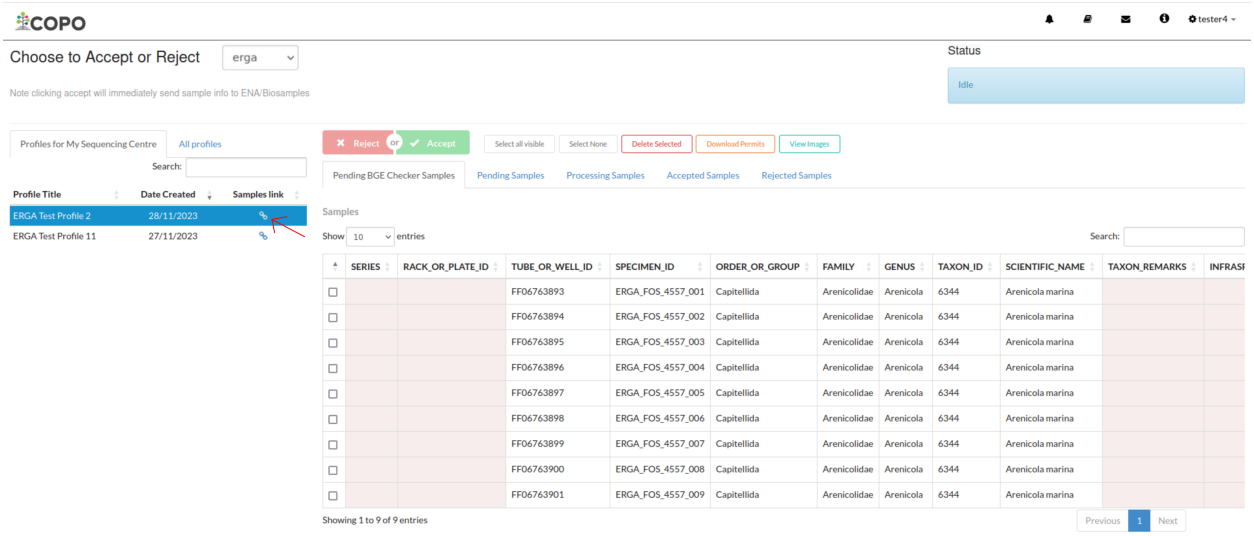


Fig. 19: **Accept or Reject Samples web page: Navigate to 'Samples' web page by clicking the link associated with an ERGA profile**

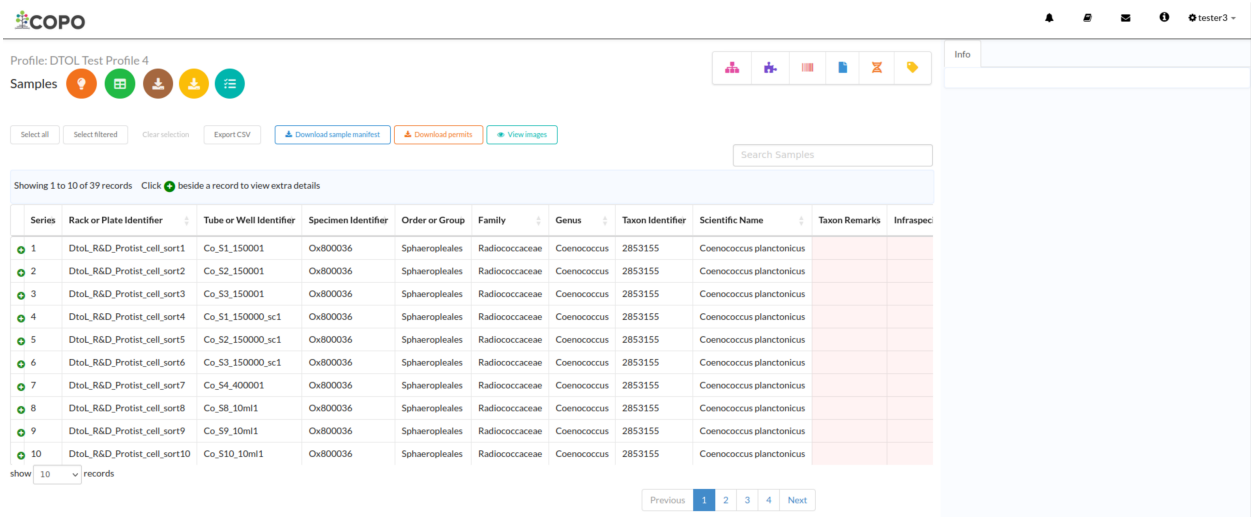


Fig. 20: **Samples web page for a DTOL profile**

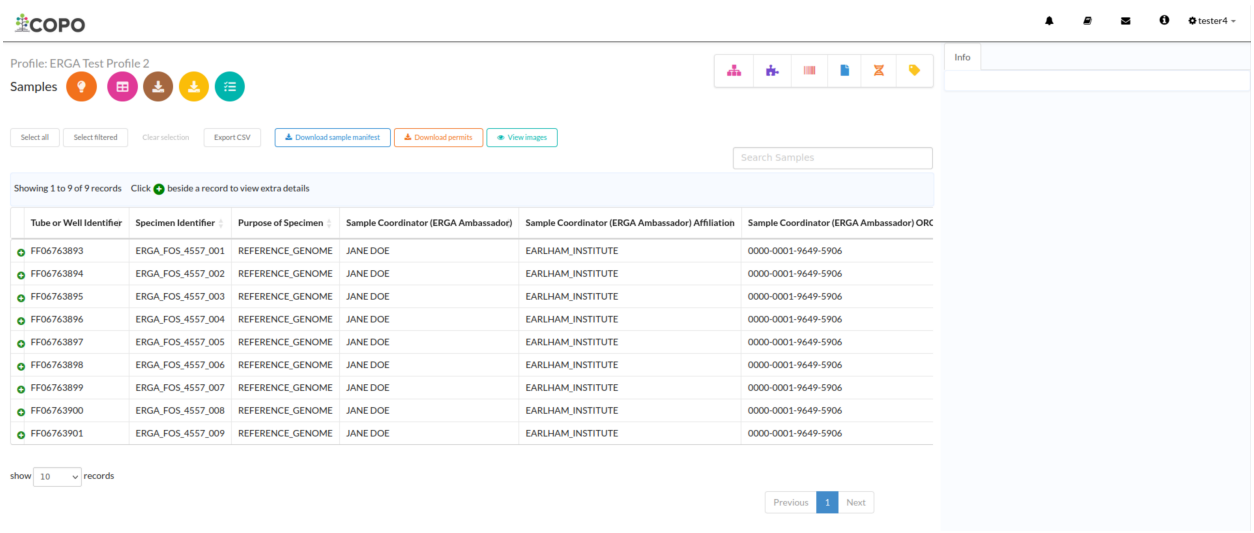


Fig. 21: **Samples web page for an ERGA profile**

The samples has to be removed from the profile so that the sample metadata in the newer version of the manifest can be registered in the profile.

Important: Please request that the samples be removed from the profile **only** if you are certain and have the newer version of the manifest with the sample metadata information.

The samples **cannot** be restored after they have been removed from the profile.

---

4. Upload the newer version of the manifest by referring to the guidelines described in the desired link below to learn more about each type of manifest submission:
    - Aquatic Symbiosis Genomics (ASG) manifest submission
    - Darwin Tree of Life (DToL) manifest submission
    - Darwin Tree of Life Environmental (DToL_ENV) manifest submission
    - European Reference Genome Atlas (ERGA) manifest submission

---

Hint: See *Downloading Manifest Templates* section for information about downloading manifest templates.

CHAPTER

FIVE

STAND-ALONE PROFILE

In COPO, a Stand-alone work profile[1] is required to submit files, reads, assemblies and sequence annotations.

See also:

- See *Steps to create Tree of Life (ToL) profile* if you would like to make ToL manifest submissions

5.1 Steps to Create a Stand-alone Profile

1. Click the **Add new record** icon to view the **Add Profile** form

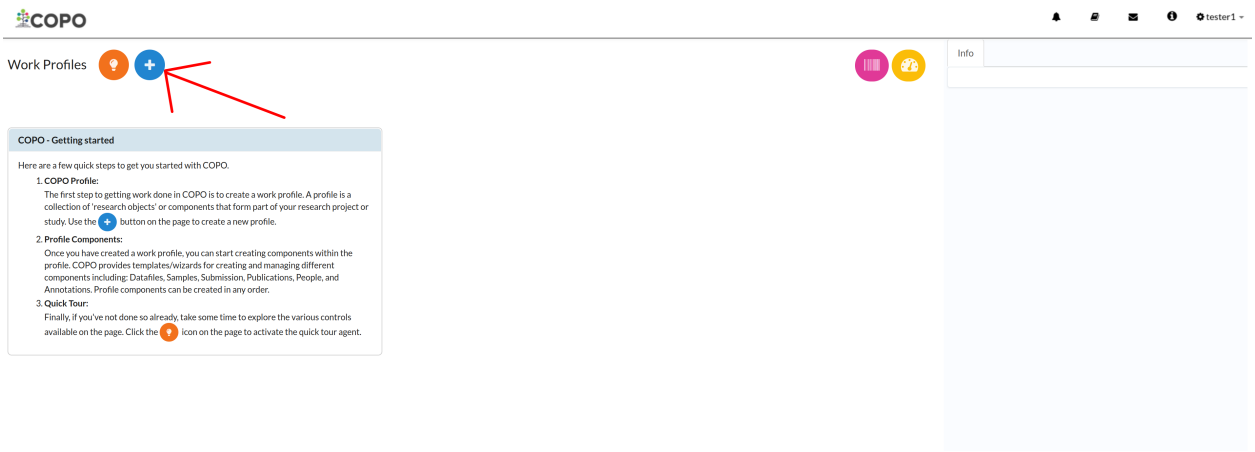


Fig. 1: **Stand-alone Profile: Add new profile record icon**

2. **Contact COPO via email** dialogue is displayed indicating that the user is not a member of any manifest group and that the user must make a request to be added to a manifest group to make ToL manifest submissions if the user would like to do so.

   Click **Okay** to close the dialogue.

> **Note:** Submissions such as files, reads, assemblies, files and sequence annotations can only be made using a Stand-alone profile type.

[1] Also known as COPO profile. See: *COPO profile or work profile*.

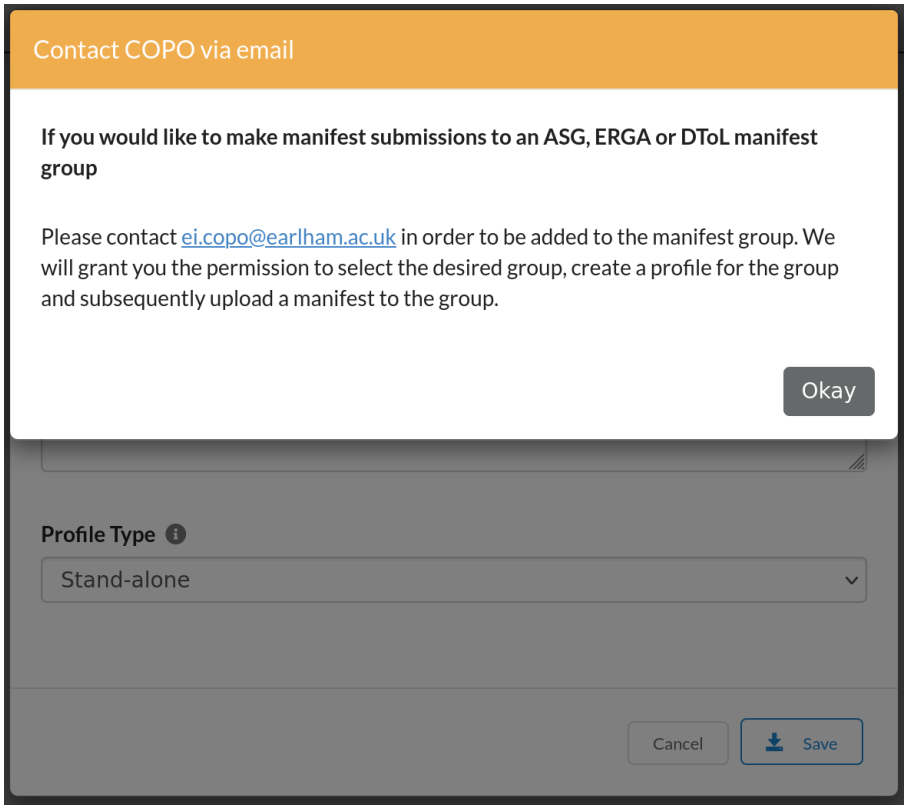


Fig. 2: **Stand-alone Profile: Contact COPO dialogue is displayed regarding getting access to make ToL manifest submissions**

3. Provide details for the new profile then, click **Save**

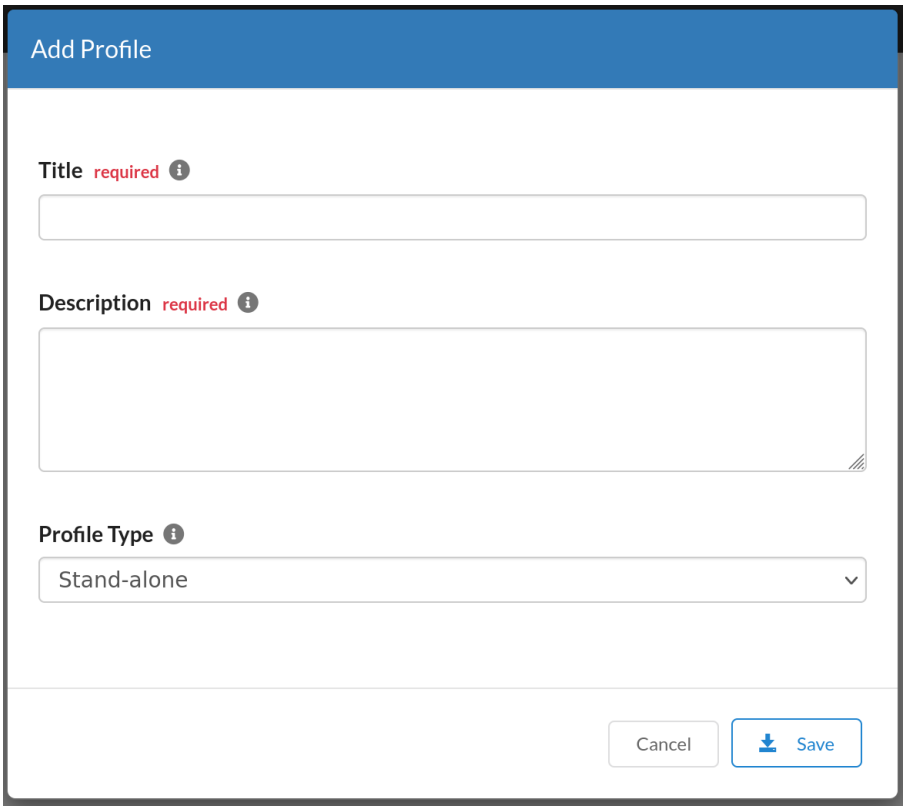


Fig. 3: **Stand-alone Profile: Add profile form dialogue**

Hint: Both profile **Title** and profile **Description** are mandatory form fields.

Meaningful field values are recommended in the form boxes because the information will appear in submissions of the research objects[2] associated with the profile, in public remote repositories.

---

4. The new profile will be displayed in the **Profile** list

Hint: The **Work Profiles**' list can be sorted by date created, profile title or profile type.

Choose the desired sort type from the **Sort by** dropdown menu (at the top-right of the profile record).

---

[2] Research objects refer to files, reads, assemblies, files and sequence annotations. A Stand-alone profile is considered as a *project* research object. See: *Profile component*.

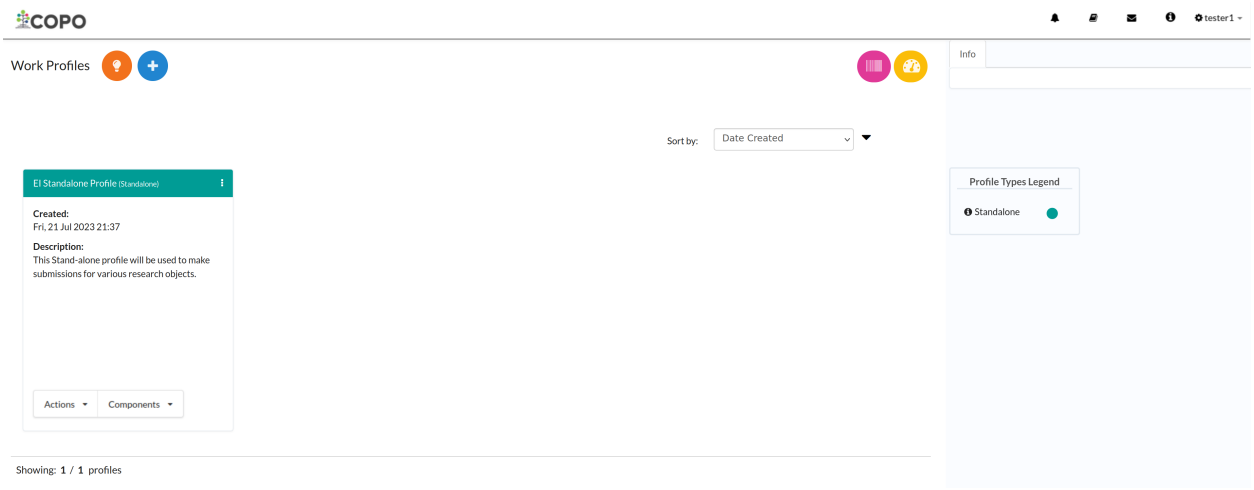


Fig. 4: **Stand-alone Profile: Work profiles' web page displaying the created profile**

5.2 Stand-alone Profile Components

A COPO profile defines a set of component types from which instances of research objects can be created.

The following component types are currently defined:

1. Accessions
2. *Assembly*
3. *Files*
4. *Reads*
5. *Sequence Annotations*

- Component instances defined within a profile will only be visible within that profile[Page 27, 1].
- To access a component within a profile, click the component button displayed within the popup after the

button was clicked (see the screenshot above).

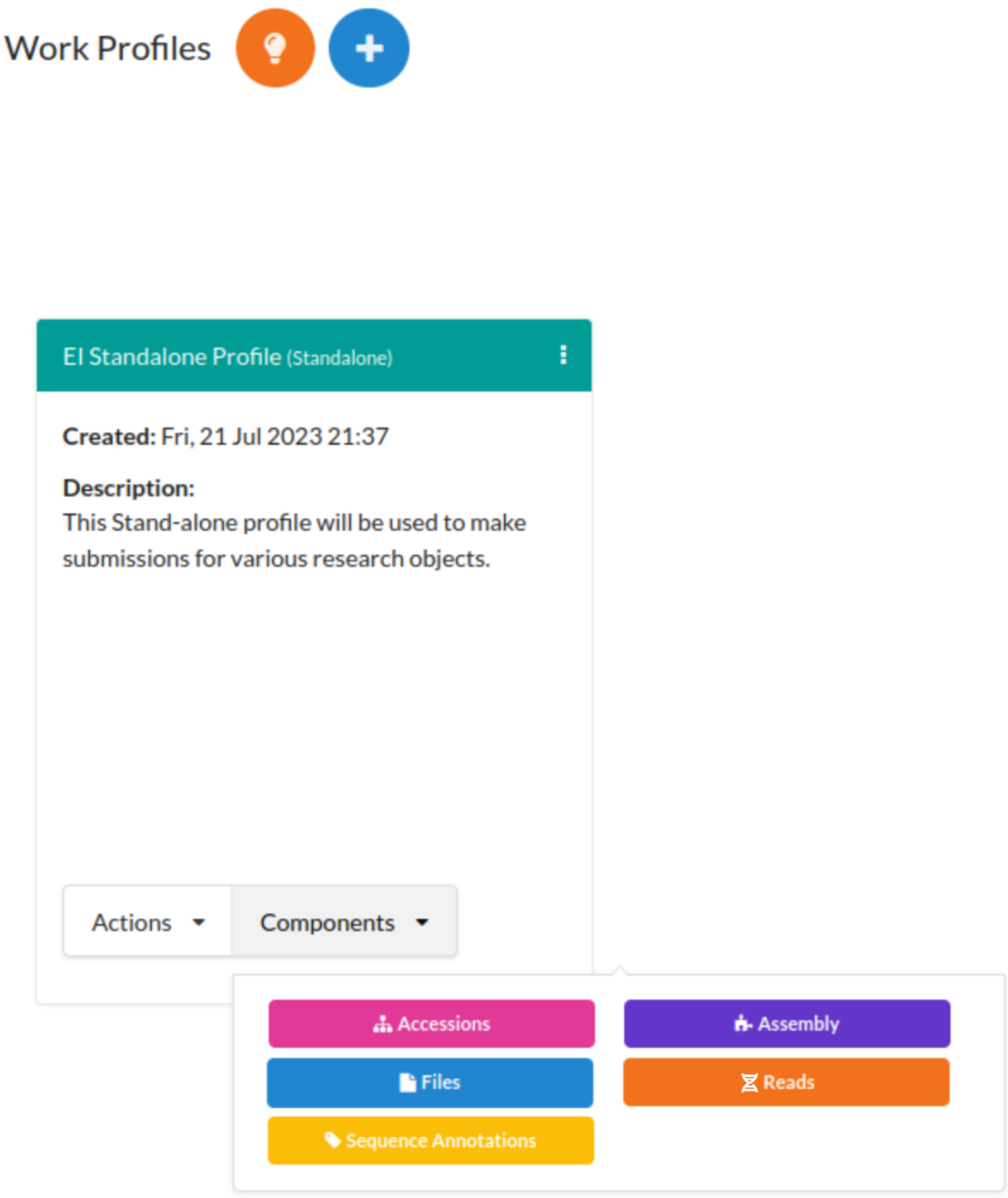


Fig. 5: **Stand-alone Profile Components**

CHAPTER

SIX

TREE OF LIFE PROFILE

In COPO, a Tree of Life (ToL)[1] work profile[2] is required to submit research objects[3] such as samples, barcoding manifests, sequence annotations, reads, assemblies and files.

A ToL profile can be used to create one of the following biodiversity projects:

- Aquatic Symbiosis Genomics (ASG)
- European Reference Genome Atlas (ERGA)
- Darwin Tree of Life Environmental (DTOL_ENV)
- Darwin Tree of Life (DTOL)

See also:

- *How to Update Profiles*
- *How to Delete Profiles*
- *Sharing Profiles*
- *Sorting Profiles*
- *Profile Types Legend*

6.1 Steps to Create a Tree of Life Profile

1. Click the **Add new record** icon to view the **Add Profile** form

2. **Contact COPO via email** dialogue is displayed indicating that the user is not a member of any manifest group and that the user must make a request to be added to a manifest group to make ToL manifest submissions if the user would like to do so.

   Click **Okay** to close the dialogue.

[1] See term: *Tree of Life (ToL)*.
[2] Also known as COPO profile. See term: *COPO profile*.
[3] Research objects refer to files, reads, assemblies, files and sequence annotations. A Stand-alone profile is considered as a *project* research object. See term: *Profile component*.

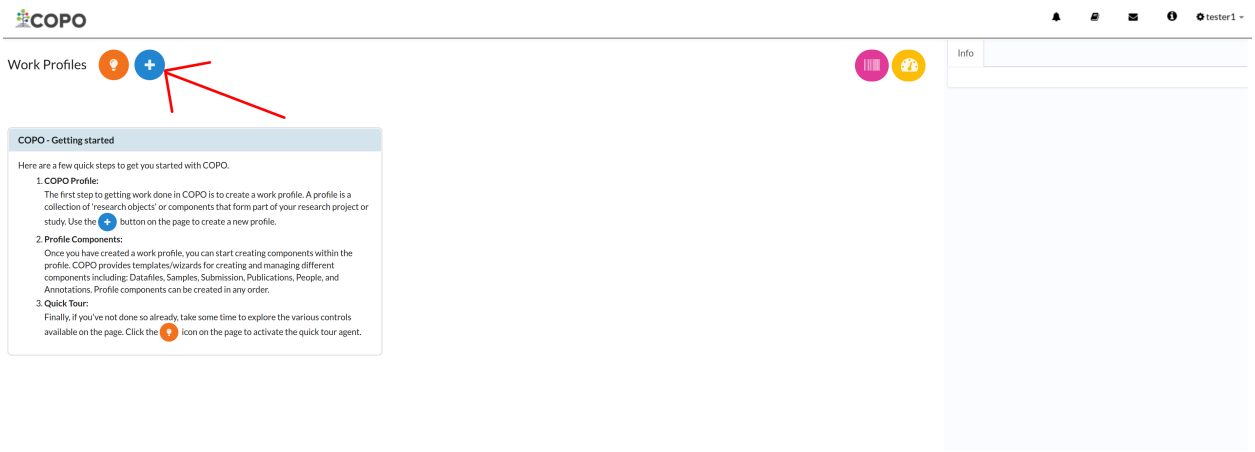


Fig. 1: **ToL Profile: Add new profile record icon**

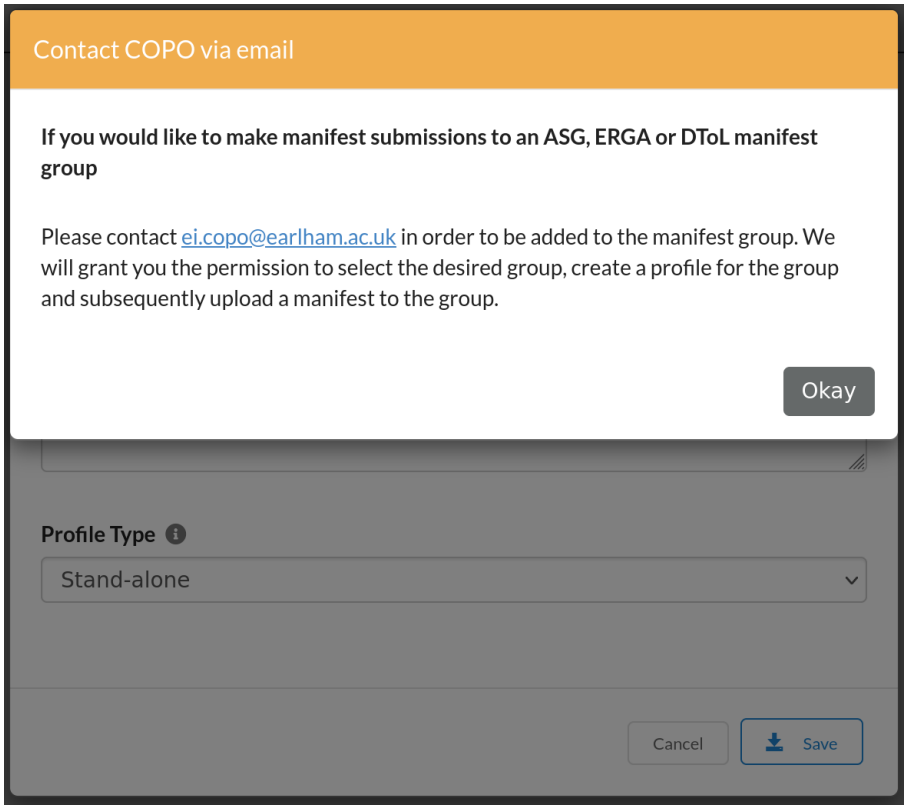


Fig. 2: **ToL Profile: Contact COPO dialogue is displayed regarding getting access to make ToL manifest submissions**

6.1.1 Provide Profile Title & Description

3. In the **Add Profile** form dialogue, provide details for the new profile then, click **Save**

Note: The **Profile Type** dropdown menu by default will only display the **Stand-alone** option.

If your desired profile type is not displayed in the **Profile Type** dropdown menu, indicating the type of profile group that you would like to be granted access to. Thereafter, the desired profile type option will be displayed in the dropdown menu.

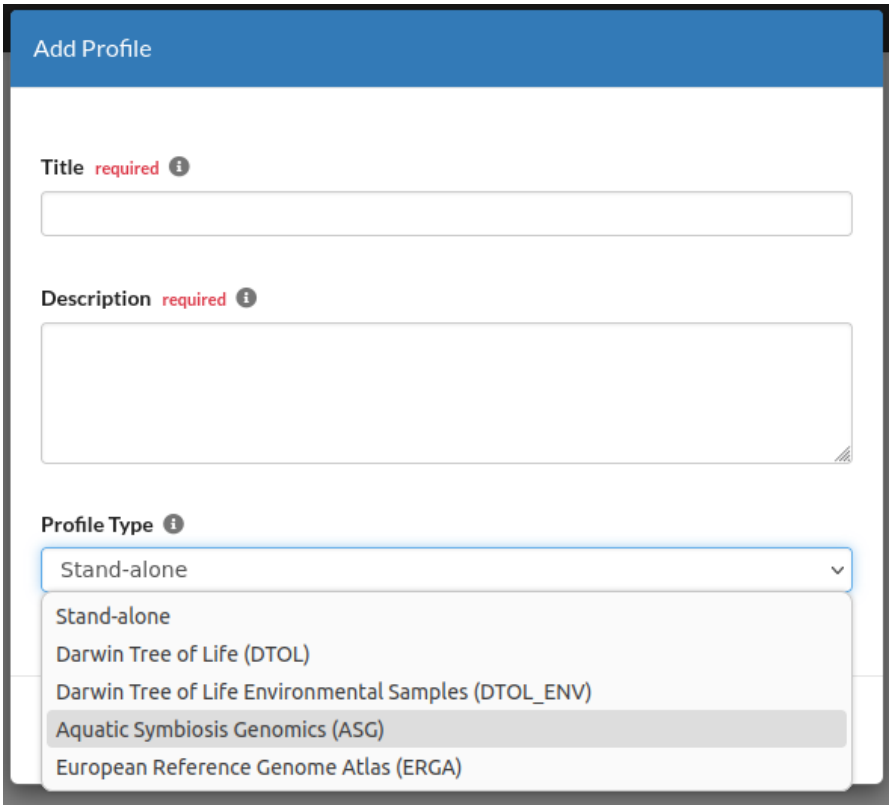


Fig. 3: **ToL Profile: Add profile form dialogue**

Hint: Both profile **Title** and profile **Description** are mandatory form fields.

Meaningful field values are recommended in the form boxes because the information will appear in submissions of the research objects associated with the profile, in public remote repositories.

6.1.2 Choose Profile Type

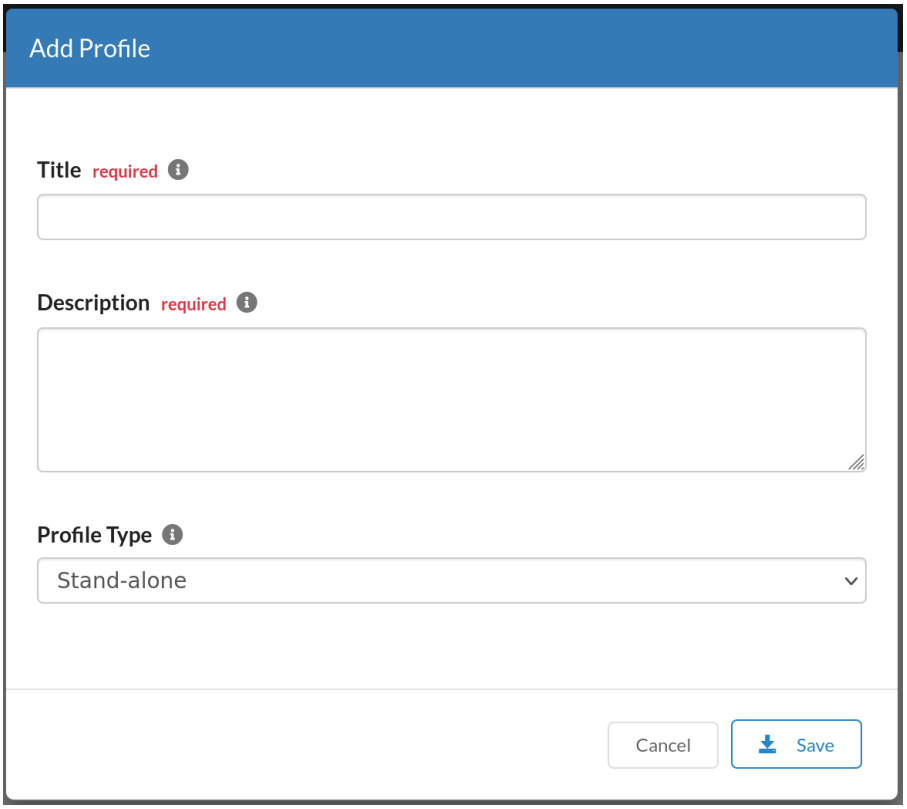


Fig. 4: **ToL Profile form: Choose profile type**

If you request to be added to any of the profile types displayed (excluding the default **Stand-alone** profile type), only that profile type will be listed in the **Profile type** dropdown menu.

6.1.3 Choose Associated Profile Type(s) (if required)

Note: The **Associated Profile Type** dropdown menu will only display if the **European Reference Genome Atlas (ERGA)** profile type is selected.

- More than one associated type also known as subproject or child project can be chosen.

OR

OR

[4] See term: *ASG*.
[5] See term: *DToL*.
[6] See term: *ERGA*.

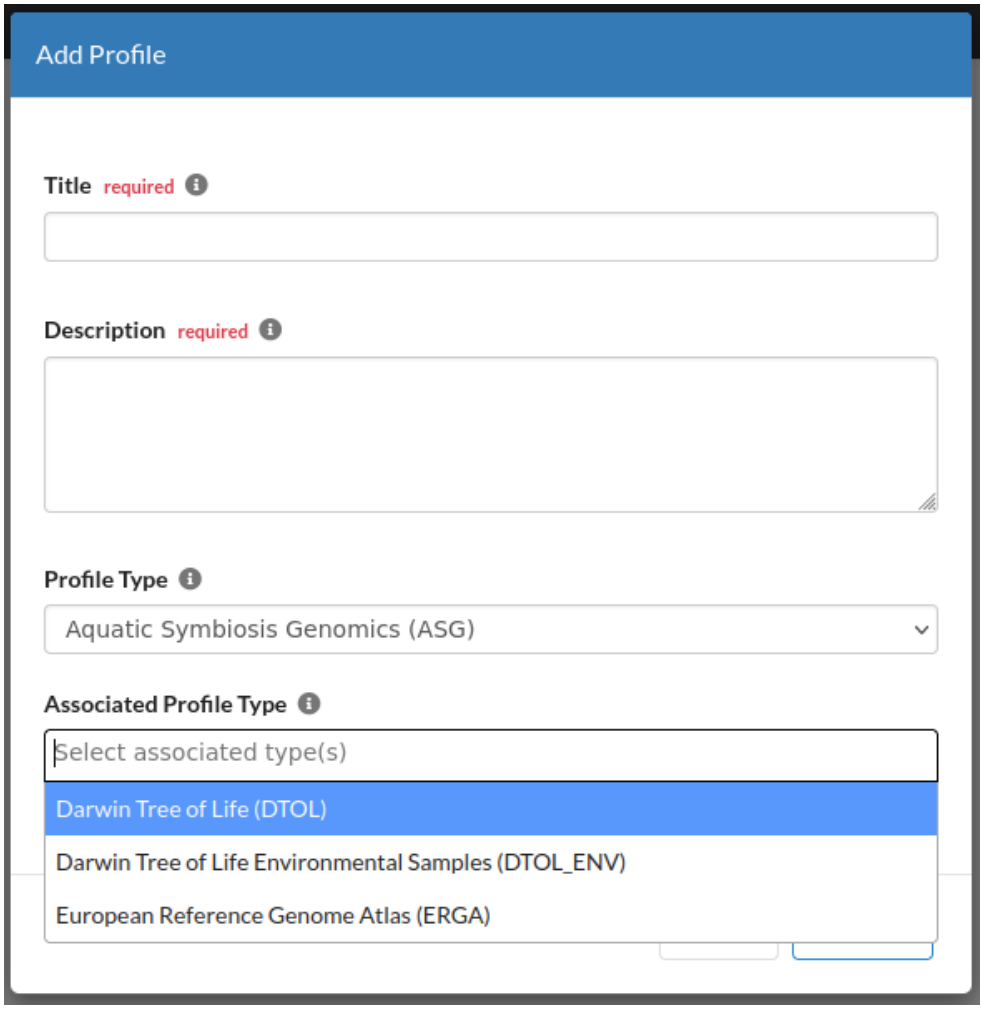


Fig. 5: **ASG Profile Type: Choose associated profile type or a subproject**[Page 36, 4]

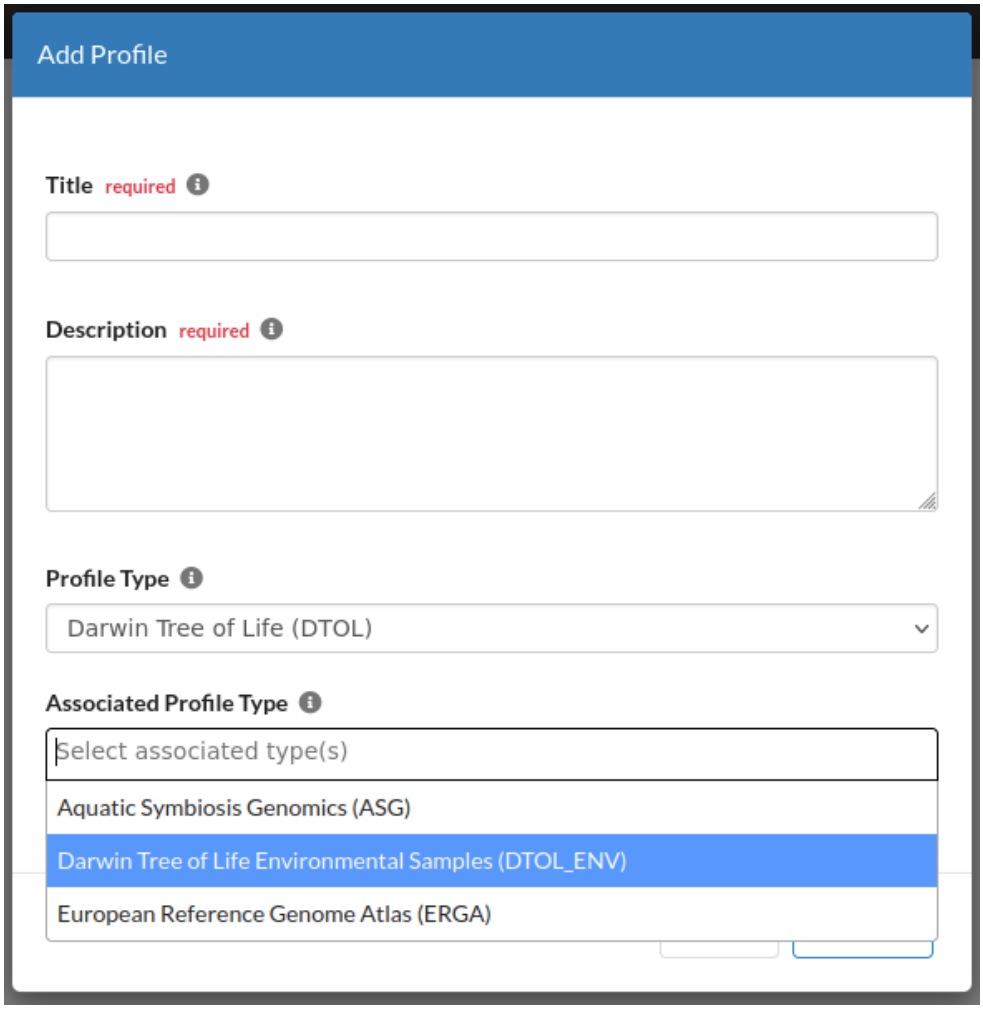


Fig. 6: **DTOL Profile Type: Choose associated profile type or a subproject**[Page 36, 5]

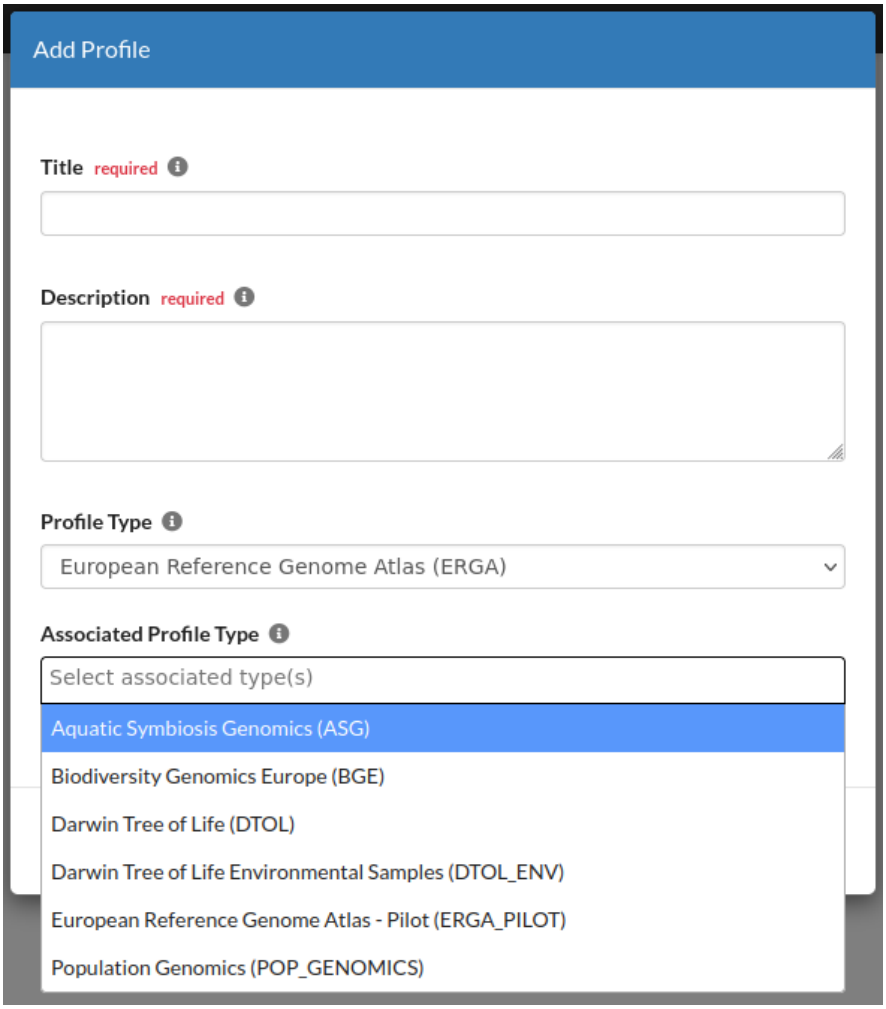


Fig. 7: **ERGA Profile Type: Choose associated profile type or a subproject**[Page 36, 6]

6.1.4 Choose Sequencing Centre (if required)

Note: The **Sequencing Centre** dropdown menu will only display if the **European Reference Genome Atlas (ERGA)** profile type is selected.

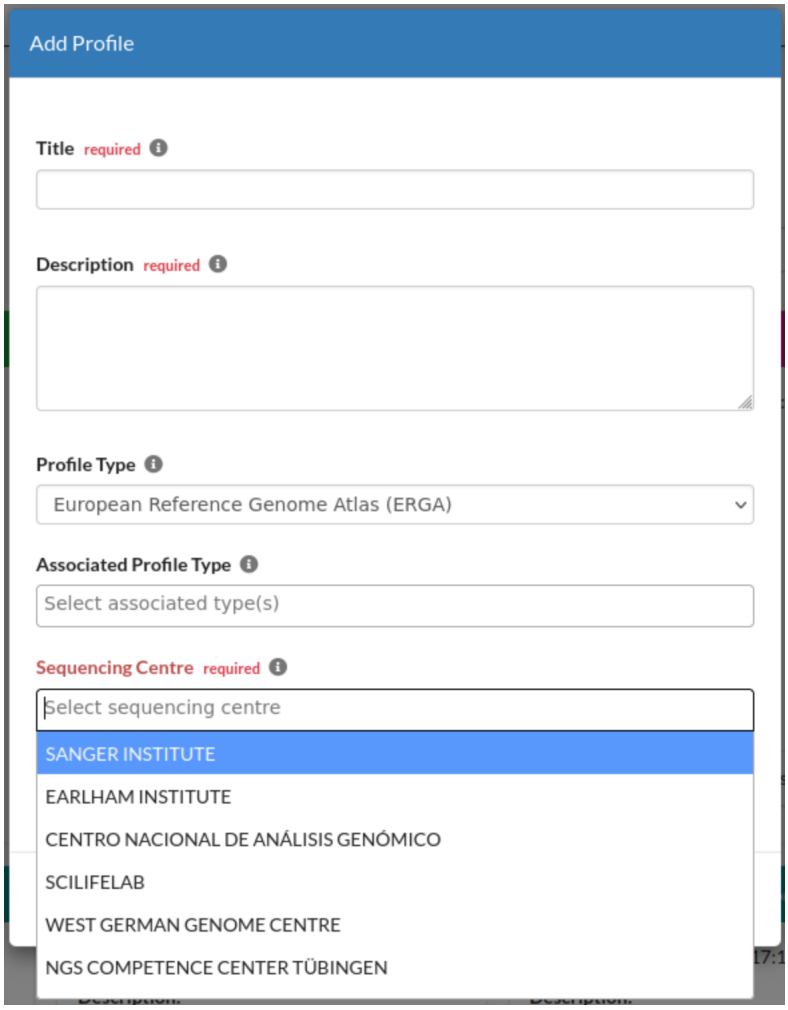


Fig. 8: **ERGA Profile Type: Choose sequencing centre**

6.1.5 Profile created

4. The new profile will be displayed in the **Profile** list

Hint: The list of profiles or profile records on the **Work Profiles** web page can be sorted by date created, profile title or profile type.

Choose the desired sort type from the **Sort by** dropdown menu (at the top-right of the profile record).

See more information in the *Sorting Profiles* section.

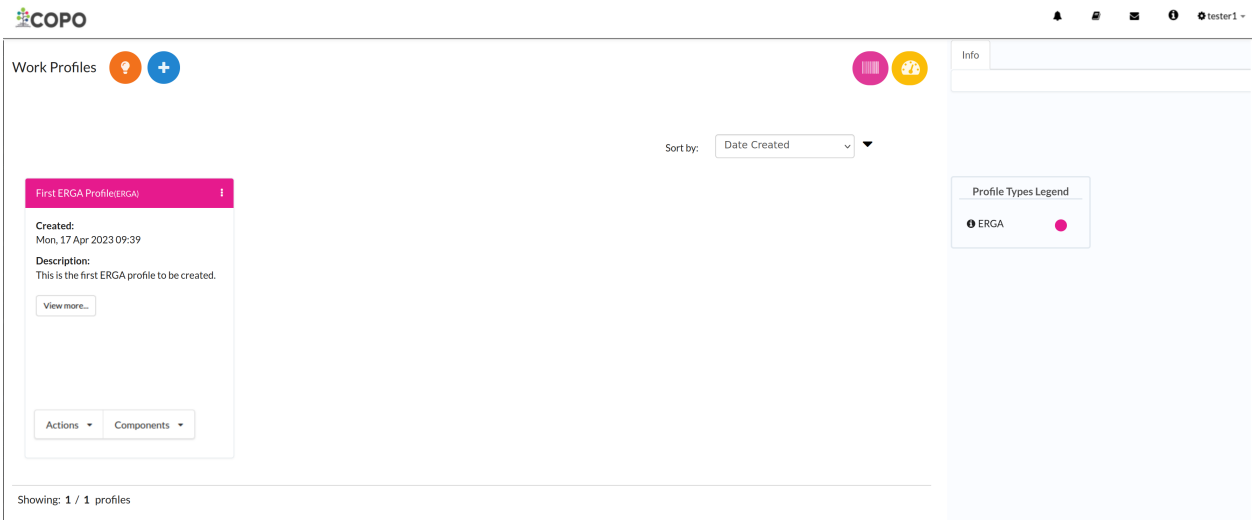


Fig. 9: **ToL Profile: 'Work Profiles' web page displaying the created profile**

See also:

- See *Steps to create Stand-alone profile* if you would like to make other submissions

6.2 Tree of Life Profile Components

A COPO profile[Page 33, 2] defines a set of component types from which instances of research objects can be created.

The following component types are currently defined:

1. Accessions
2. *Assembly*
3. Barcoding manifests
4. *Files*
5. *Reads*
6. Samples
7. *Sequence Annotations*

- Component instances defined within a profile will only be visible within that profile.
- To access a component within a profile, click the component button displayed within the popup after the Components button was clicked (see the screenshot above).

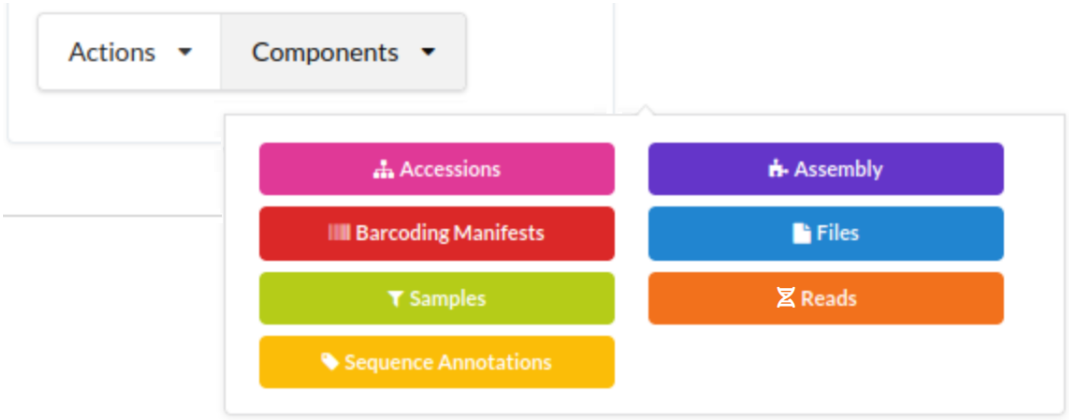


Fig. 10: **Tree of Life Profile Components**

CHAPTER

SEVEN

SHARING PROFILES

Sharing profiles or creating a group enable other COPO users to upload or submit research objects such as samples, experiment, runs, reads, assemblies and sequence annotations etc. to a profile owned or created by another COPO user.

Hint:

- **Profile owner** or **Sharer** refers to the person who created the profile.
- **Shared user** or **Sharee** refers to the person who has been granted permission by the profile owner to upload or submit research objects to the profile created by the profile owner.

In general, the **sharer** shall create a profile, create a group, add the profile to be shared to the created group then, add the COPO user(s) to the created group.

Thereafter, the shared profile shall appear on the **Work Profiles** web page of the **sharee**.

The following describes how to share a profile or create a group on COPO:

7.1 Sharer

1. Click your name in the top right hand corner of the web page (after you have logged in) to access settings
2. Click the **View Groups** option from the settings dropdown menu as shown below to navigate to the **Groups** web page

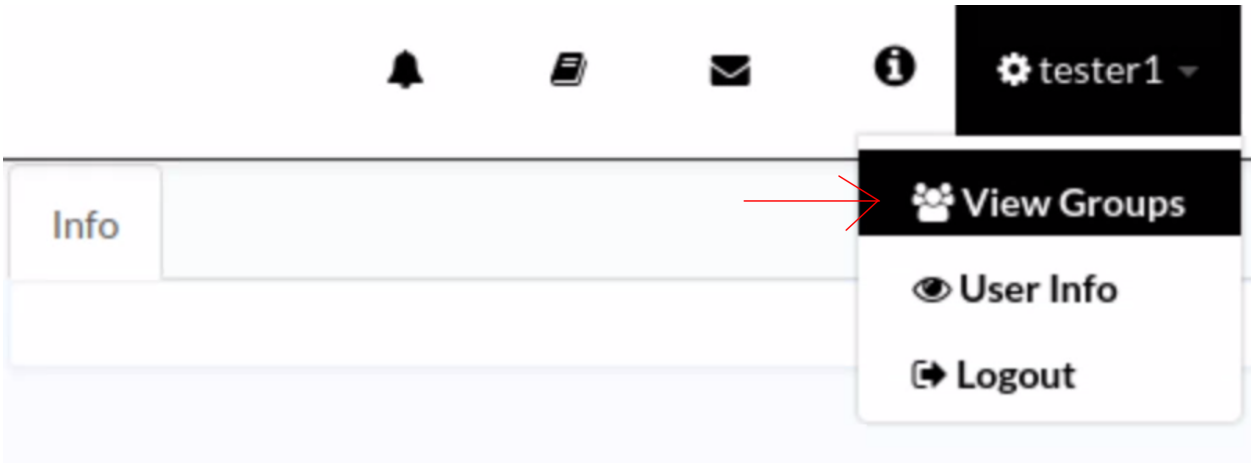


Fig. 1: **Groups: Navigating to Groups web page via the 'View Groups' settings dropdown menu option**

3. Click the button to create a new group

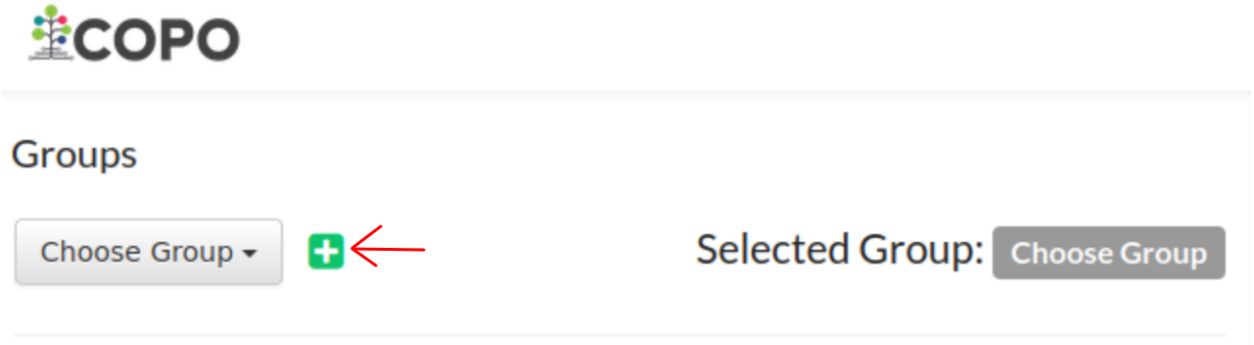


Fig. 2: **Groups: Groups web page**

4. Provide details for the new group as shown below then, click **Submit**

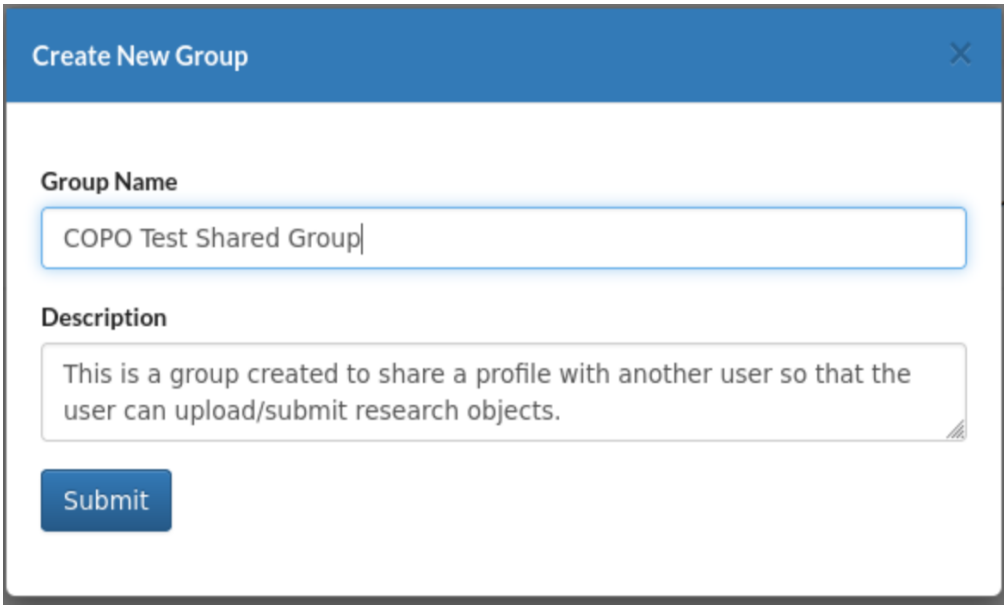


Fig. 3: **Groups: Add new group form dialogue**

Hint: Both group **Name** and profile **Description** are required form fields.

Meaningful field values are recommended in the form boxes.

---

5. In the **Your Profiles** section under the **Profiles in Group** header, choose a profile from the list of profiles that you would like to share with another user (so that the user can upload/submit research objects to) by **double-clicking** the profile title.

   The profile is then, transferred to the **Added to Group** section after the profile title was double-clicked as shown below.

6. Search for the user who you would like to share the profile with in the **Users in Group** section located at the right of the web page. Then, click the name of the user to add the user to the group.

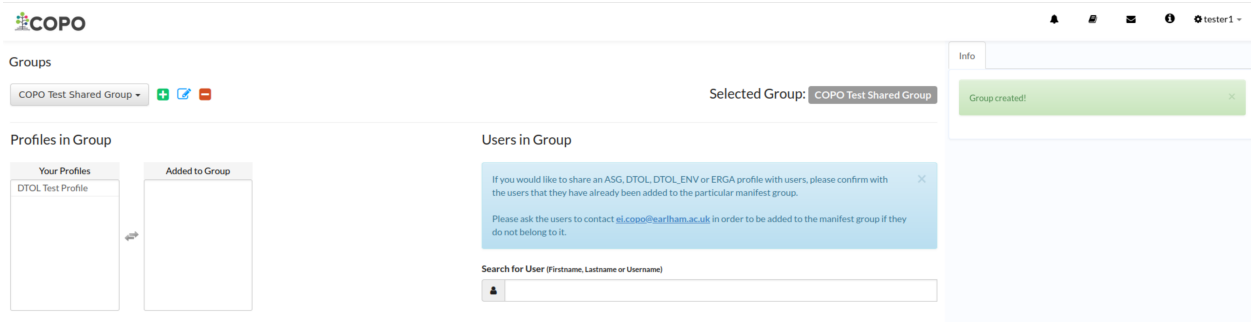


Fig. 4: **Groups: Group created successfully**

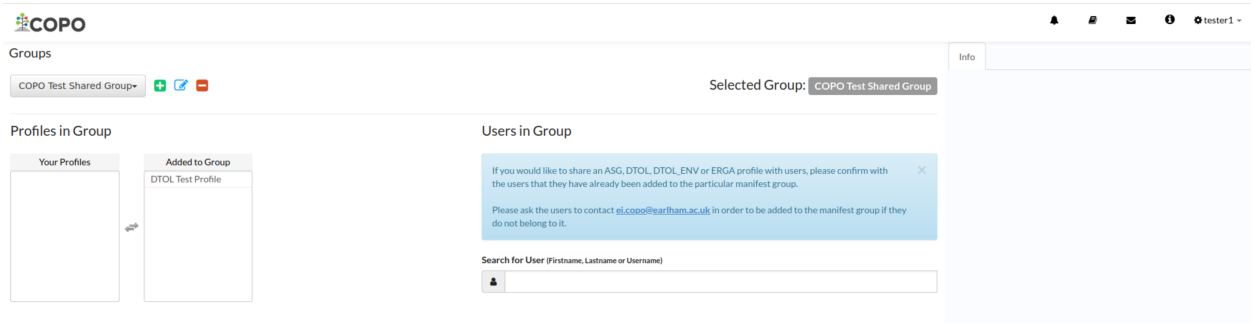


Fig. 5: **Groups: Profile appears in 'Added to Group' section after having been double-clicked in the 'Your Profiles' section**

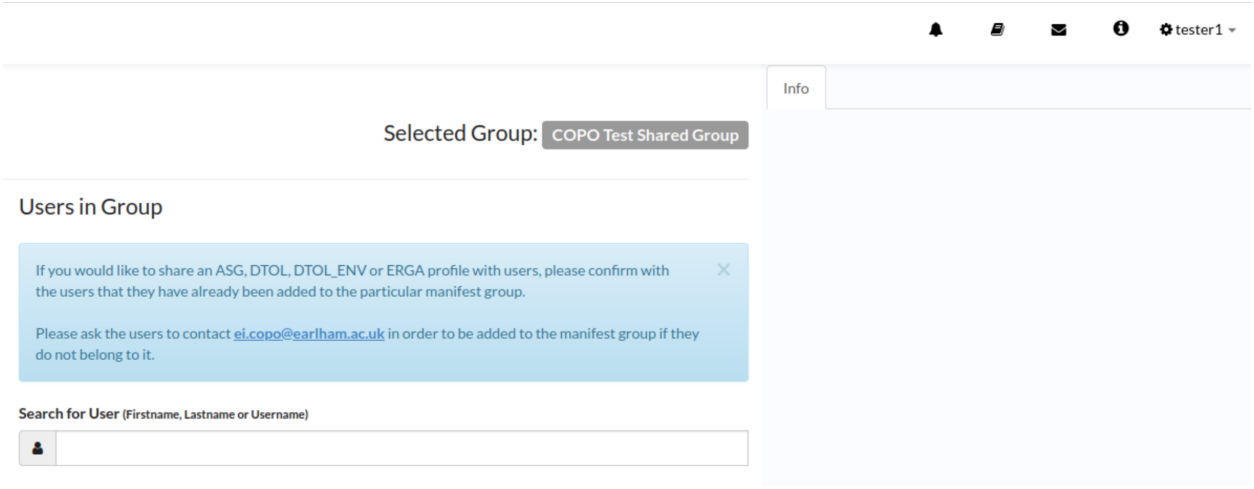


Fig. 6: **Groups: Use search box to search for users to add to the created group**

7.2 Sharee

1. Login to COPO.
2. Request to be added to a profile group associated with that the profile type that you would like to be added if you would like to submit or upload research objects to a Tree of Life (ToL) project.

   If you have not been added to the profile group, make a request to the indicating the type of profile group that you would like to be assigned to.

   See the *Projects Brokered through COPO* section for information about types of ToL projects.
3. After the sharer has added you to the group, you will see the shared profile on the **Work Profiles** web page as shown below:

Note:

- If the profile owner has shared the profile with you and you do not see the shared profile on the **Work Profiles** then, please perform step 2.
- Shared profiles are indicated by an orange label as shown below.

  See *Profile Types Legend* section for more information about profile types' colour labels.

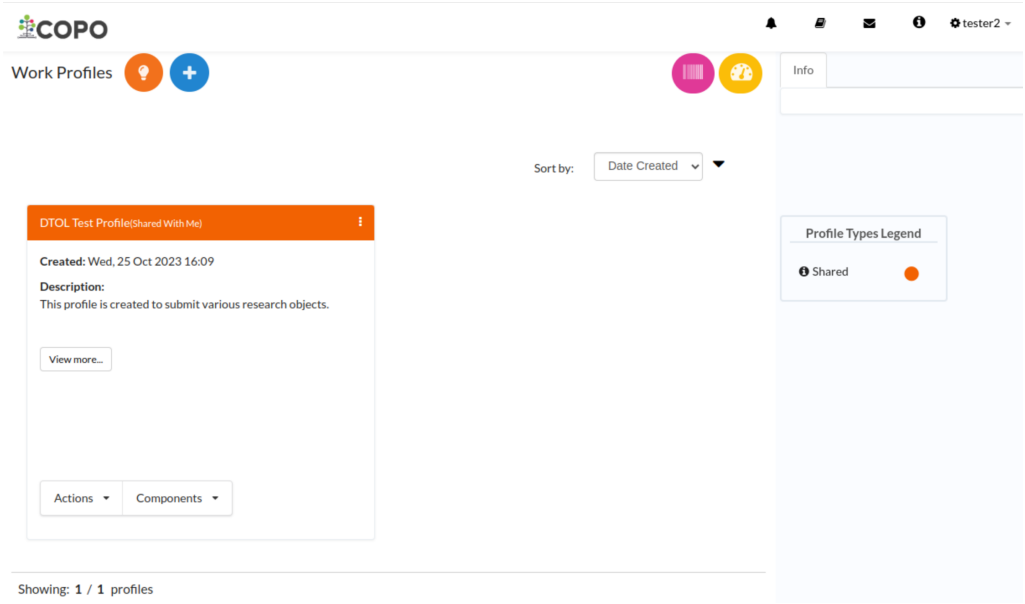


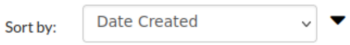
Fig. 7: **Groups: Shared profile appears on the sharee's 'Work Profiles' web page**

CHAPTER

EIGHT

SORTING PROFILES

On the **Work Profiles** web page, profiles[1] can be sorted by **date created**, **profile title** or **profile type**.

Choose the desired sort option from the **Sort by** dropdown menu, 

, (at the top-right of the profile record section) as shown below.

Then, click the ▼ arrow to sort in descending order or the ▲ arrow to sort in ascending according to the chosen sort option.

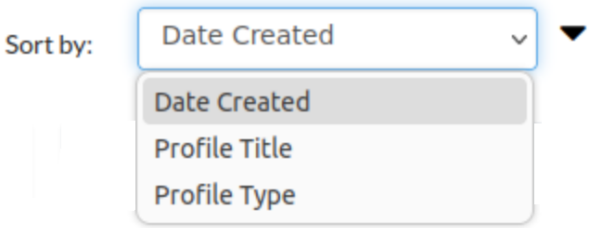


Fig. 1: **Profile Sort Menu: Sort options displayed**

[1] Also known as COPO profile. See term: *COPO profile*.

CHAPTER

NINE

PROFILE TYPES LEGEND

The **Profile Types legend**[1] is displayed at the right of the **Work Profiles** web page.

It displays the acronym as well as the corresponding profile type colour for each profile that has been created.

Hint: Hover over the ℹ icon located beside the profile type acronym in the **Profile Types Legend**, the backronym for the profile type will be displayed.

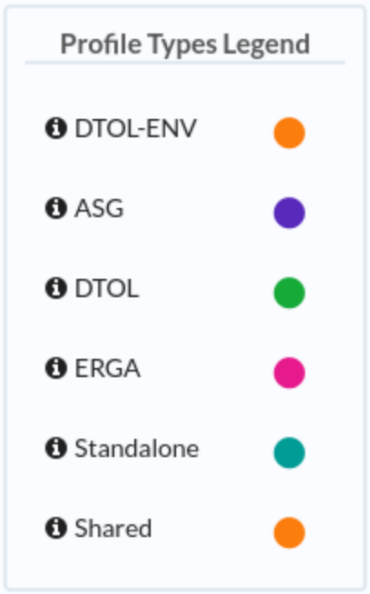


Fig. 1: **Profile Types Legend: Various profiles types that can be created in COPO and their corresponding label colours**

[1] See term: *Profile Types legend*.

Table 1: Explanation of Profile Types Legend

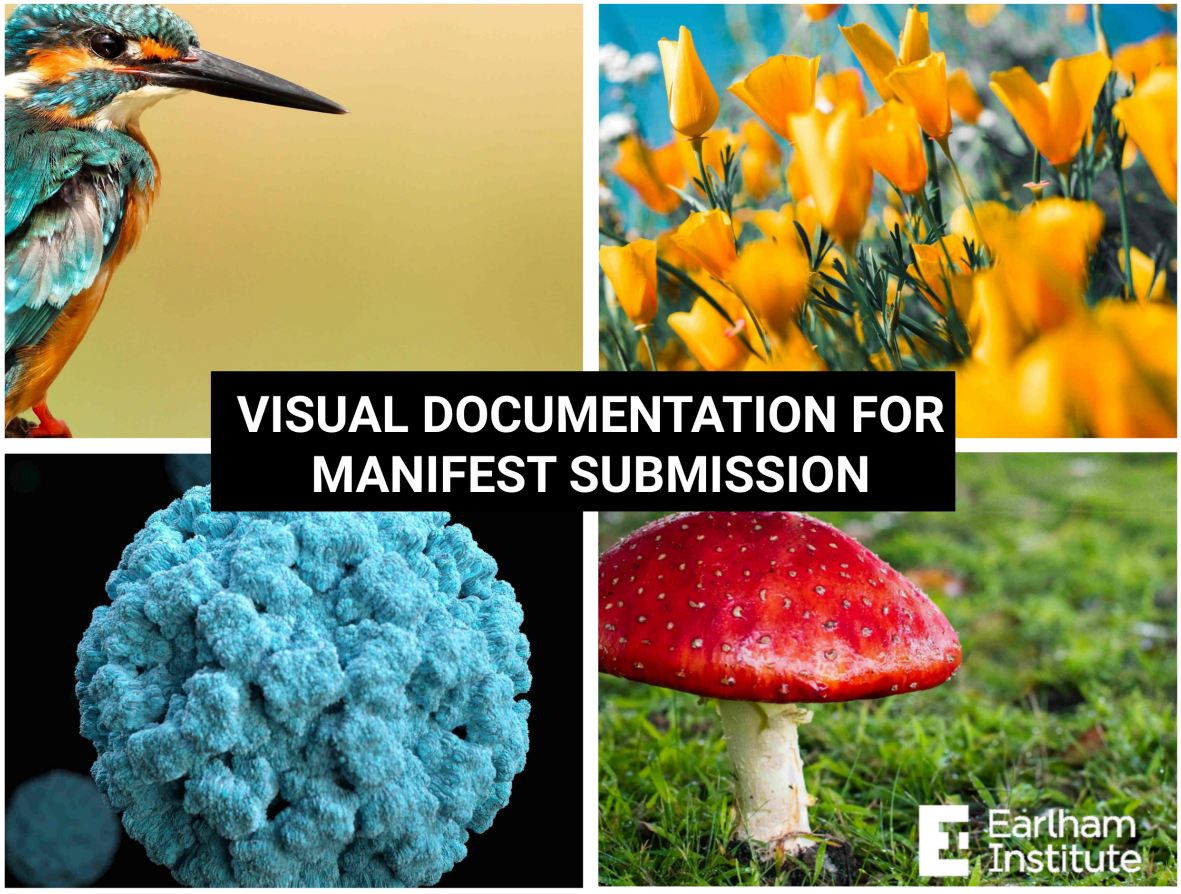
Colour	Acronym	Backronym
Orange	DTOL-ENV **or** Shared	Darwin Tree of Life Environmental **or** Shared
Purple	ASG	Aquatic Symbiosis Genomics
Green	DTOL	Darwin Tree of Life
Pink	ERGA	European Reference Genome Atlas
Teal	Standalone	Stand-alone

CHAPTER

TEN

GENERAL TREE OF LIFE VISUAL SUBMISSION DOCUMENTATION

Download visual documentation



CHAPTER

ELEVEN

ASSEMBLY SUBMISSION

Note:

- Once **assemblies** have been submitted, they cannot be updated.

See also:

- *How to Delete Assemblies*

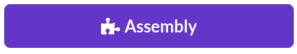
11.1 How to Submit Assemblies

11.1.1 Accessing the Assembly Web Page

The **Assembly**[1] web page can be accessed from the **Components** button or **Actions** button associated with a profile[2].

Use Components' Button to Navigate to Assembly Web Page

Click the

button associated with a profile. Then, click the 

from the popup menu displayed as shown below:

Use Actions' Button to Navigate to Assembly Web Page

Click the

button associated with a profile. Then, click the action, **Submit Assembly** action from the popup menu displayed as shown below:

[1] See: *Assembly*.
[2] Also known as COPO profile. See: *COPO profile or work profile*.

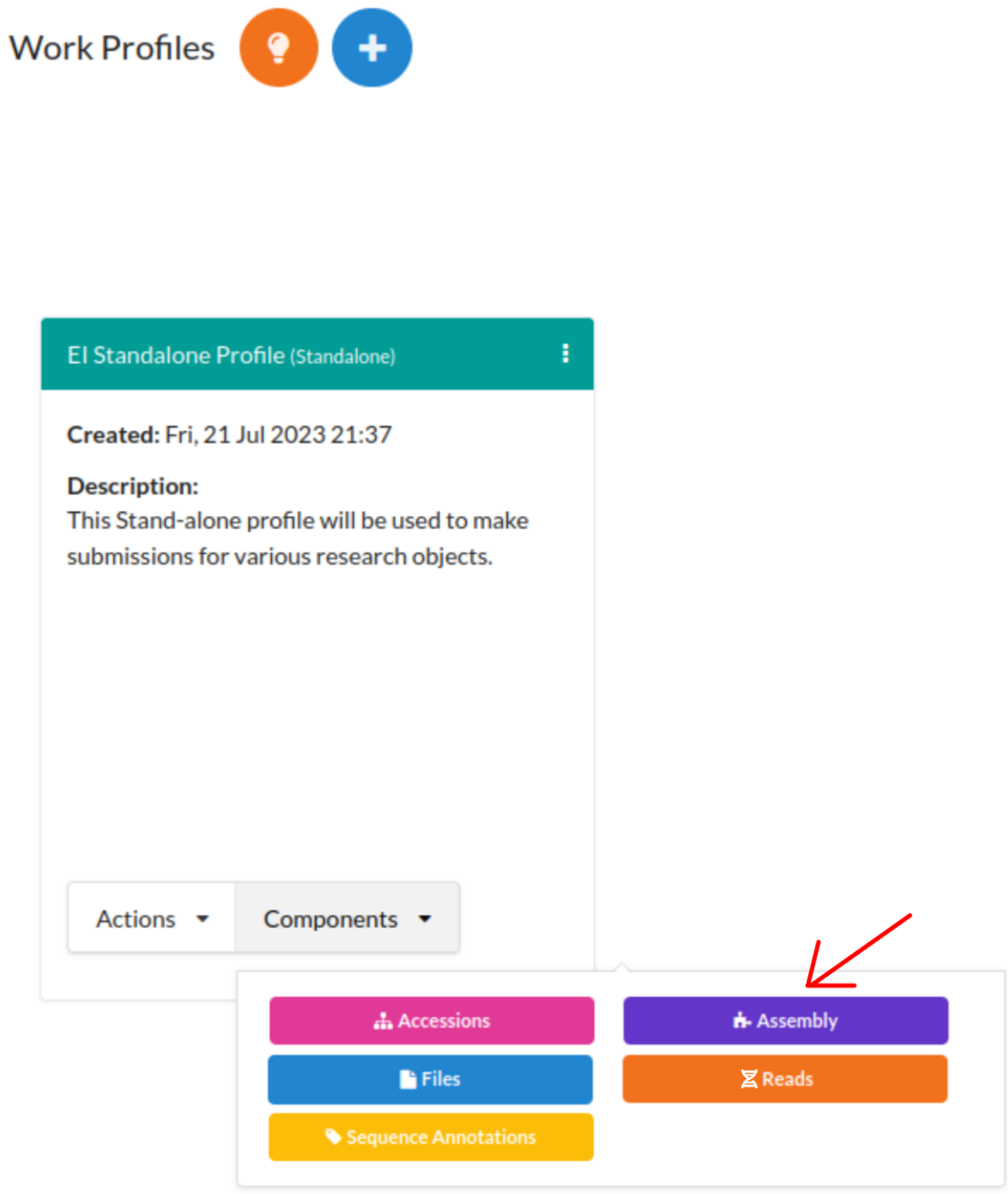


Fig. 1: **Stand-alone Profile Components: Assembly component button**

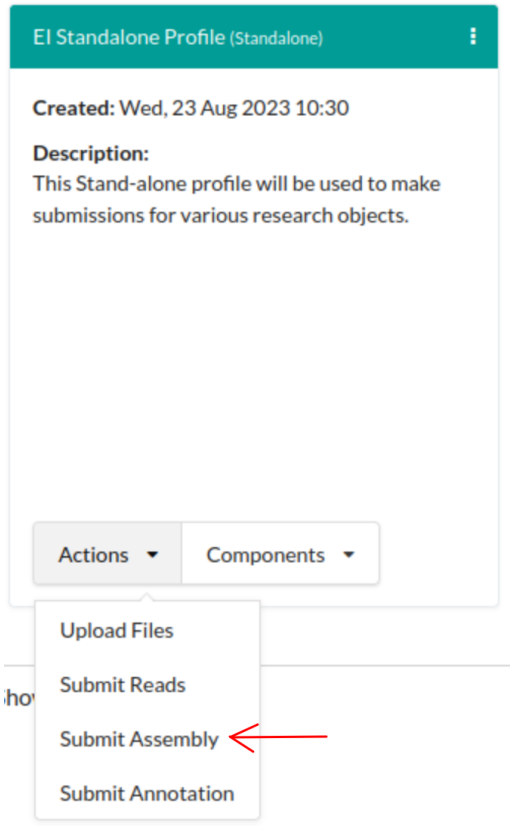


Fig. 2: **Stand-alone Profile Actions: 'Submit Assembly action**

11.1.2 Submit Assembly

1. Click button to add an **assembly** as shown below:

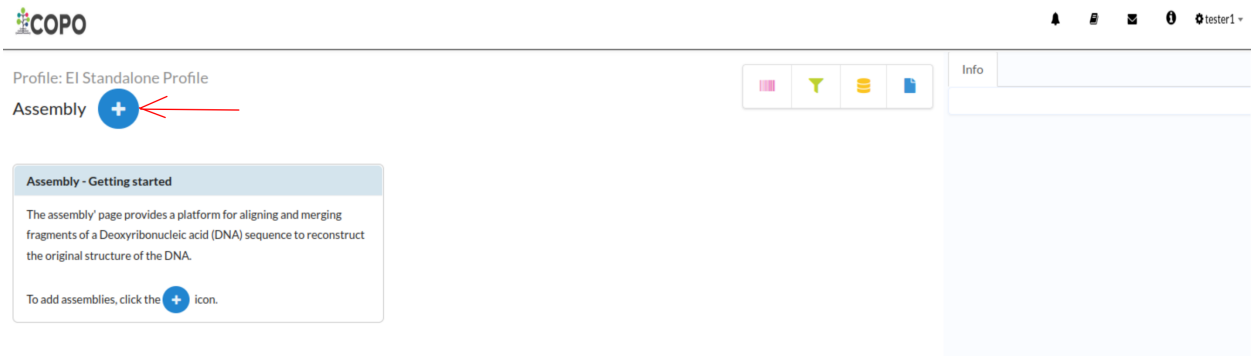


Fig. 3: **Assembly submission: Click 'Add record' button**

2. An **Add Assembly** dialogue is displayed. Provide the details then, click the **Submit Assembly** button.

3. The new assembly will be displayed on the **Assembly** web page after a successful submission.

11.2 How to Delete Assemblies

Click the desired assembly from the list of assemblies displayed on the **Assembly** web page. Then, click the **Delete** button (located in the top-right corner of the table) as shown below:

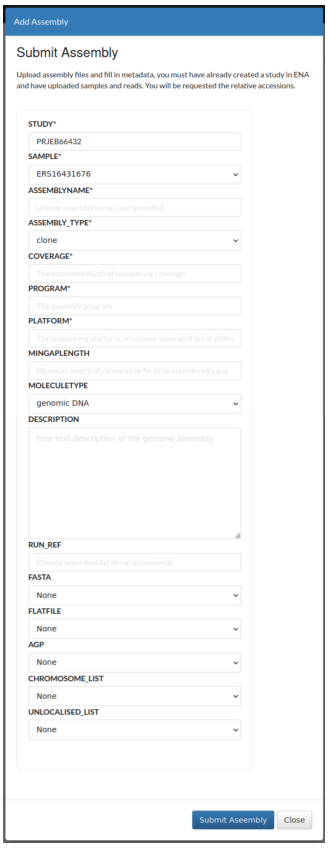


Fig. 4: **Assembly submission: 'Add Assembly' dialogue**

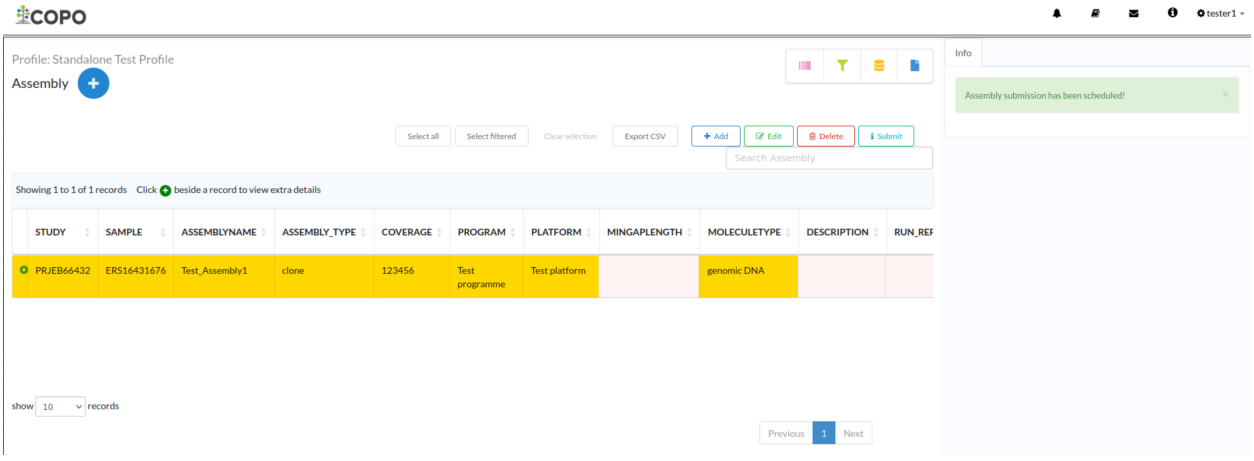


Fig. 5: **Assembly submission: Assembly web page displaying the uploaded assemblies**

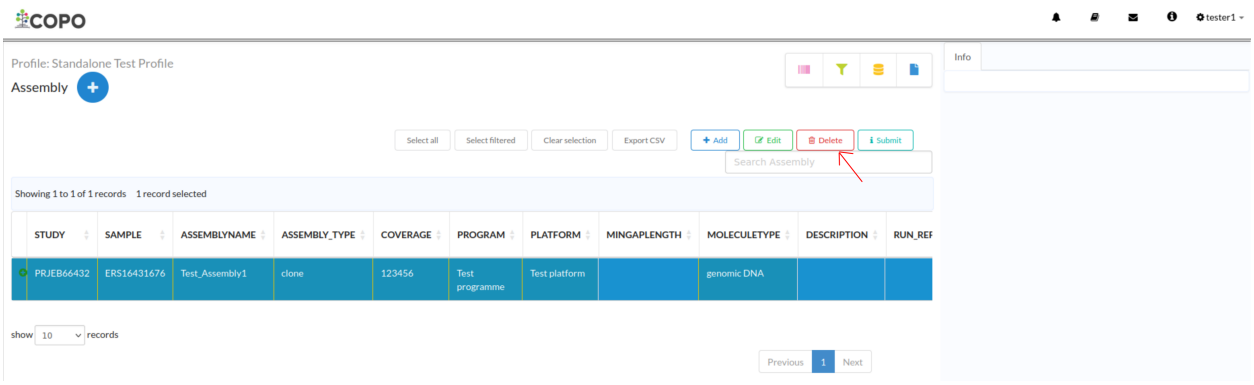


Fig. 6: **Assembly deletion: Click the "Delete" button to remove the highlighted assembly from the profile**

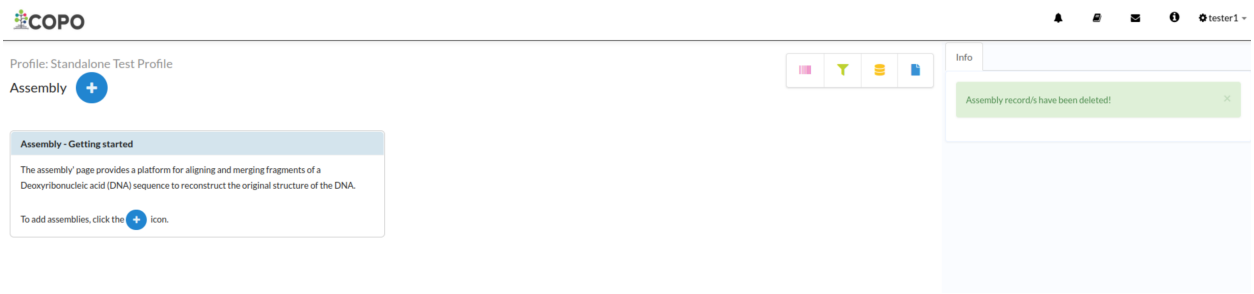


Fig. 7: **Assembly deletion: Assembly record has been deleted**

CHAPTER

TWELVE

BARCODING MANIFEST SUBMISSION

Note:

- **Barcoding manifest** submission can only be done via a ToL[1] profile[2]. Please see: *Steps to Create a Tree of Life Profile* for guidance.
- Samples must be submitted before barcoding manifests can be submitted. Once the samples have been approved their biosample accessions, specimen ID (identification) and taxon ID are needed to fill in the barcoding manifest prior submission.

See also:

- *How to Delete Barcoding Manifests*
- How to Submit Aquatic Symbiosis Genomics (ASG) samples
- How to Submit Darwin Tree of Life (DToL) samples
- How to Submit Darwin Tree of Life Environmental (DToL_ENV) samples
- How to Submit European Reference Genome Atlas (ERGA) samples

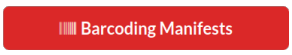
12.1 How to Submit Barcoding Manifests

12.1.1 Accessing the Barcoding Manifest Web Page

The **Barcoding Manifest** web page can be accessed from the **Components** button or **Actions** button associated with a profile.

Use Components' Button to Navigate to Barcoding Manifest Web Page

Click the

button associated with a profile. Then, click the 

from the popup menu displayed as shown below:

[1] See term: *Tree of Life (ToL)*.
[2] Also known as COPO profile. See: *COPO profile or work profile*.

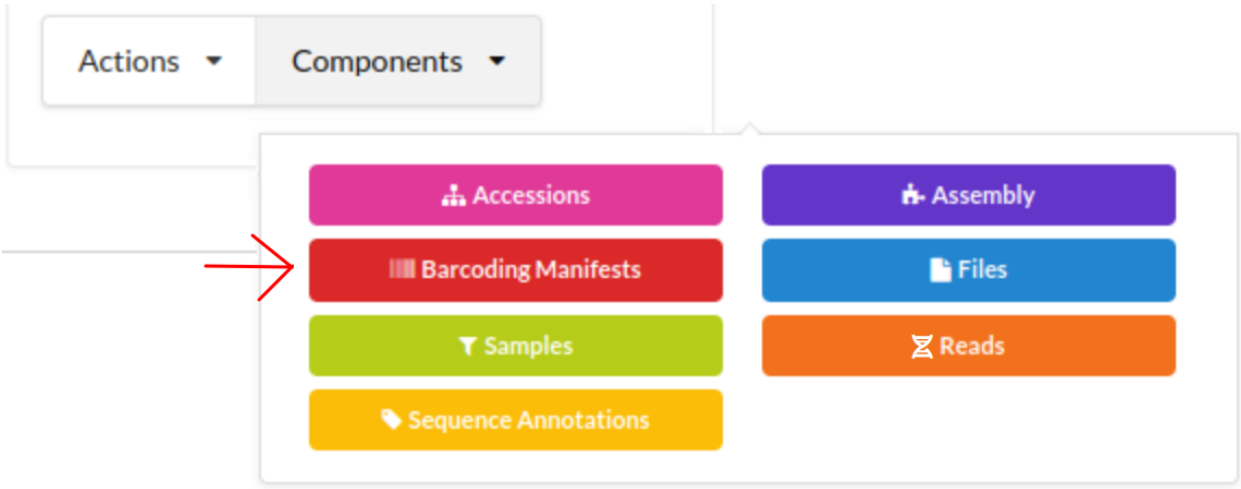


Fig. 1: **Tree of Life Profile Components: Barcoding Manifest component button**

Use Actions' Button to Navigate to Barcoding Manifest Web Page

Click the Actions button associated with a profile. Then, click the action, **Submit Barcoding Manifest** action from the popup menu displayed as shown below:

Note: The action will be available for all of the **Tree of Life** profile types.

12.1.2 Upload Barcoding Manifest

Hint: To download a blank **Barcoding Manifest** template, click the button.

---

1. On the **Barcoding Manifests'** web page, click the checklist dropdown to view a list of available checklists that support **Barcoding Manifests'** submission as shown below:

2. Click button to add **Barcoding manifest** for the chosen tagged sequence checklist as shown below:

3. An **Upload Barcoding Manifest** dialogue is displayed. Click the Upload Barcoding Manifest button to choose a file from your local system.

4. The uploaded manifest is shown in a table in the **Upload Barcoding Manifest** dialogue as shown below. Click the Finish button to submit the barcoding manifest.

5. The new barcoding manifest(s) will be displayed on the **Barcoding Manifests** web page after a successful submission.

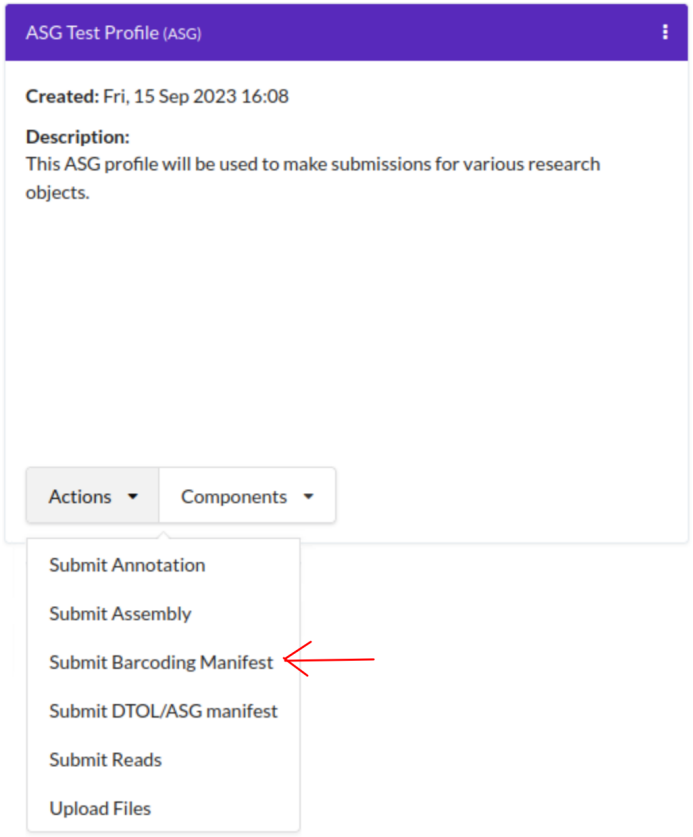


Fig. 2: **Tree of Life Profile Actions: 'Submit Barcoding Manifest action**

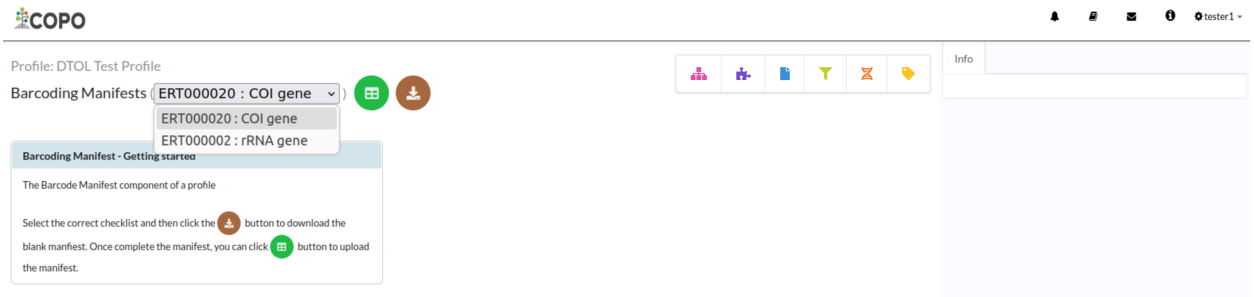


Fig. 3: **Barcoding Manifests' web page: Checklist dropdown menu with checklist options displayed**

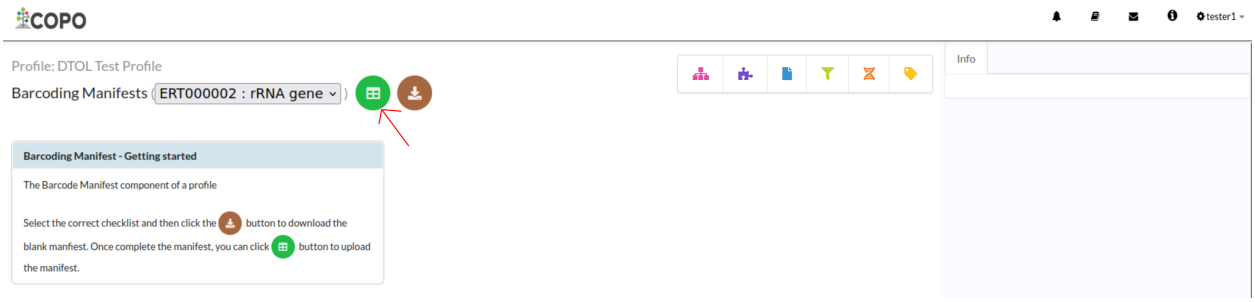


Fig. 4: **Barcoding Manifest upload: Click 'Add Tagged Sequence (s) from Tagged Sequence Spreadsheet' from Spreadsheet' button**

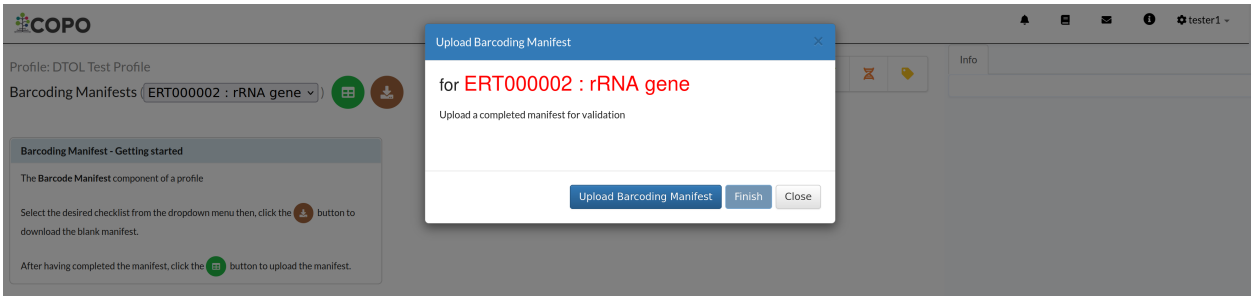


Fig. 5: **Barcoding Manifest upload: 'Upload Barcoding Manifest' dialogue**

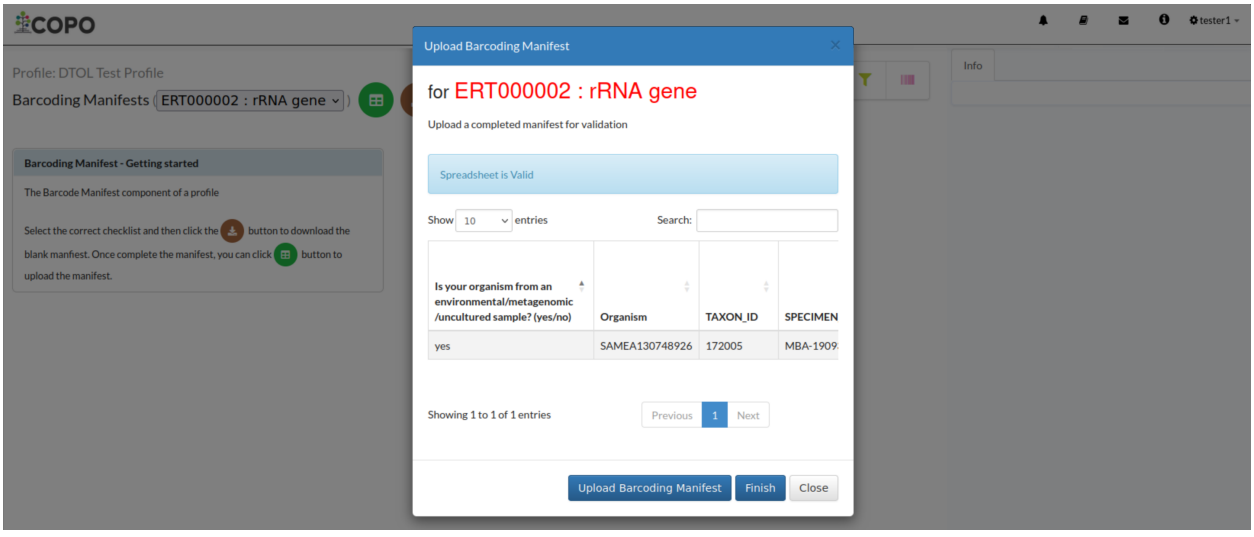


Fig. 6: **Barcoding Manifest upload: 'Upload Barcoding Manifest' dialogue with uploaded manifest**

Hint: Barcoding manifest records that are highlighted **yellow** indicate that the records are pending submission. The records will be highlighted **green** after a successful submission.

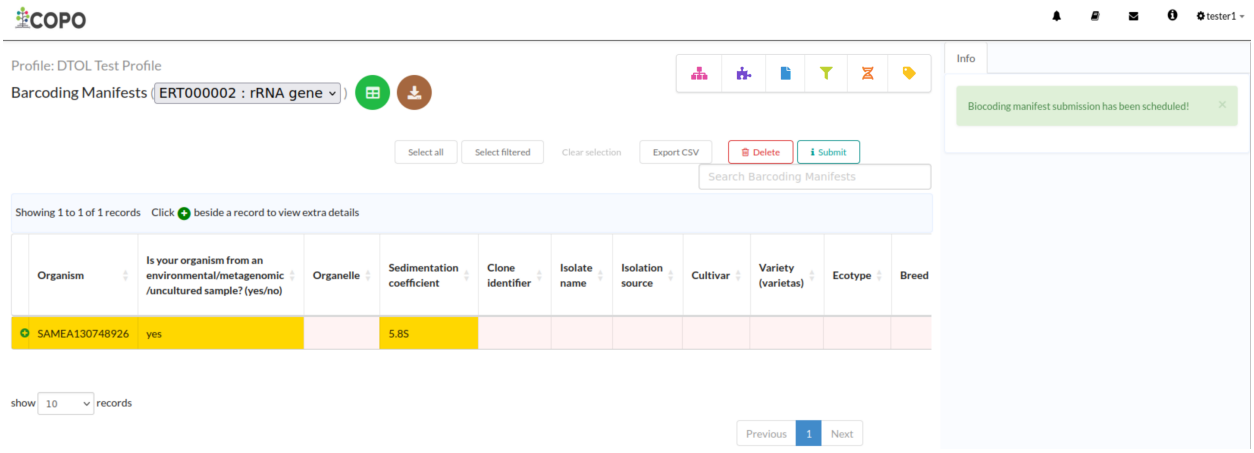


Fig. 7: **Barcoding manifest upload: Barcoding Manifests' web page displaying the uploaded barcoding manifests(s)**

12.1.3 Submit Barcoding Manifests

Hint: The submitted barcoding manifest record will be highlighted **green**.

Click the desired barcoding manifests records from the list of barcoding manifests displayed on the **Barcoding manifests** web page. Then, click the **Submit** button (located in the top-right corner of the table) as shown below:

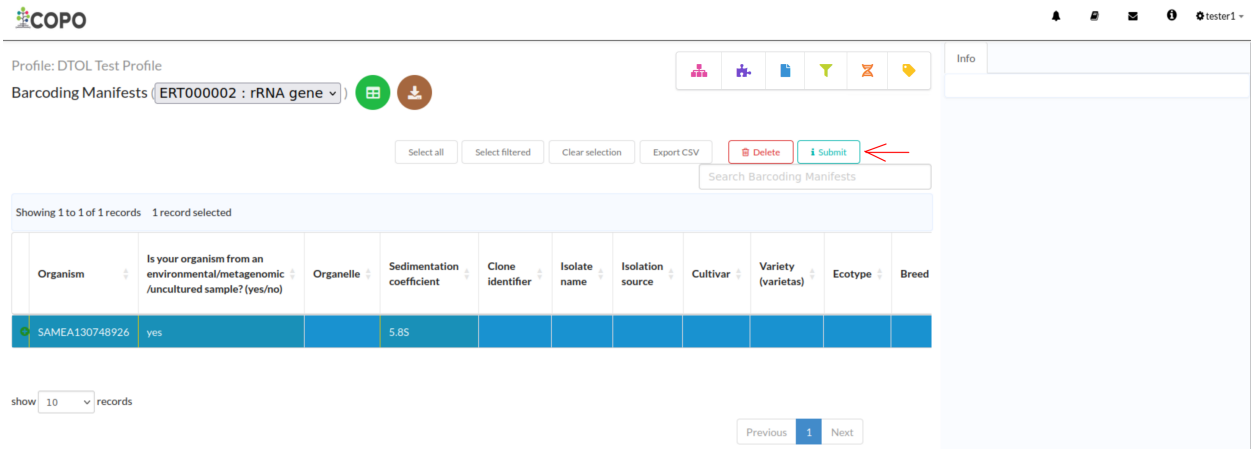


Fig. 8: **Barcoding manifest submission: Click the "Submit" button to submit the highlighted barcoding manifest from the profile**

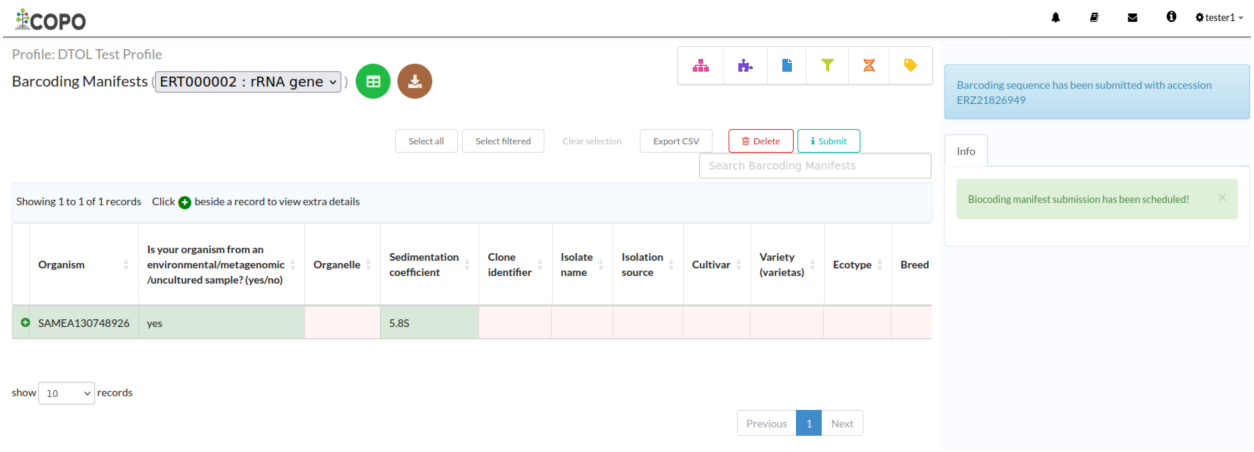


Fig. 9: **Barcoding manifest submission: The barcoding manifest has been submitted**

12.2 How to Delete Barcoding Manifests

Note: Barcoding manifests can only be deleted **before** they have been submitted.

Click the desired barcoding manifest from the list of barcoding manifests displayed on the **Barcoding Manifests** web page. Then, click the **Delete** button (located in the top-right corner of the table) as shown below:

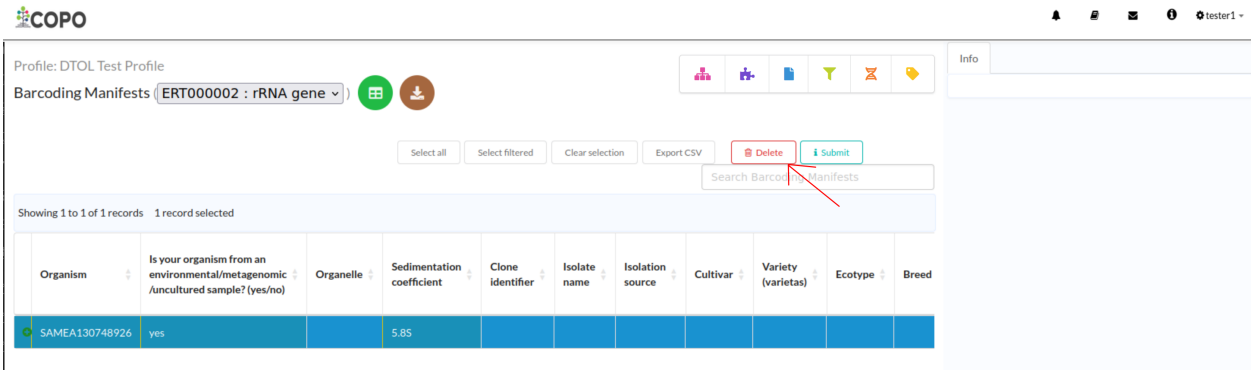


Fig. 10: **Barcoding manifest deletion: Click the "Delete" button to remove the highlighted barcoding manifest from the profile**

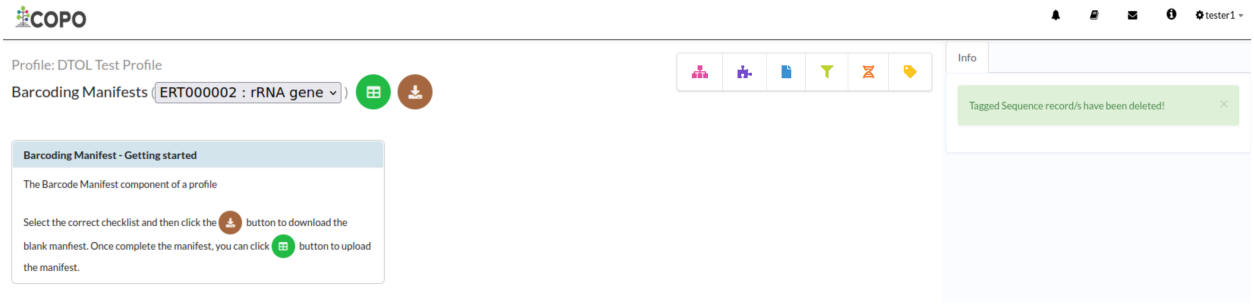


Fig. 11: **Barcoding manifest deletion: Barcoding manifest record has been deleted**

CHAPTER

THIRTEEN

FILES SUBMISSION

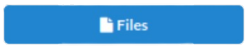
13.1 How to Submit Files

13.1.1 Accessing the Files' Web Page

The Files' web page can be accessed from the **Components** button or **Actions** button associated with a profile[1].

Use Components' Button to Navigate to Files' Web Page

Click the

button associated with a profile. Then, click the 

from the popup menu displayed as shown below:

Use Actions' Button to Navigate to Files' Web Page

Click the

button associated with a profile. Then, click the action, **Upload Files** action from the popup menu displayed as shown below:

13.1.2 Submit Files from your Local (Computer) System

1. Click the button on the Files web page to add a new file by browsing your local file system

2. An **Upload File** dialogue is displayed. Click the **Upload** button to choose a file from your local system.

3. The new file(s) will be displayed on the **Files** web page after a successful submission.

Hint: To add more files from your local system, click the

button (once files have been submitted to the profile) as an alternative to clicking the button.

---

[1] Also known as COPO profile. See: *COPO profile or work profile*.

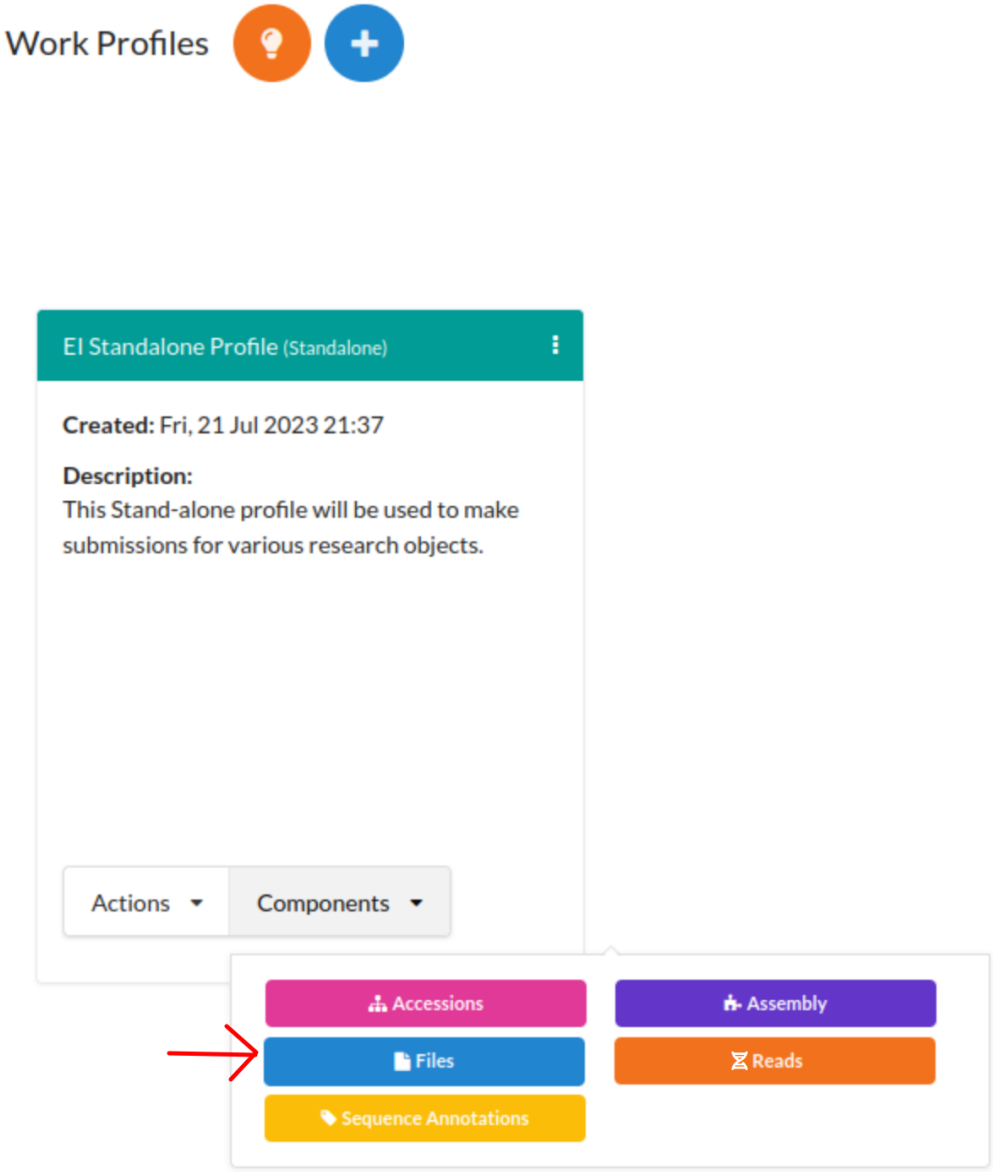


Fig. 1: **Stand-alone Profile Components: Files component button**

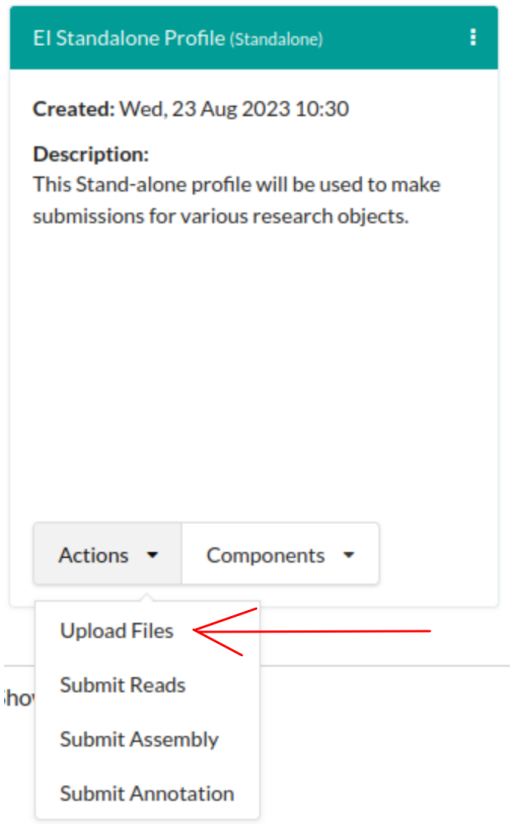


Fig. 2: **Stand-alone Profile Actions: 'Upload Files' action**

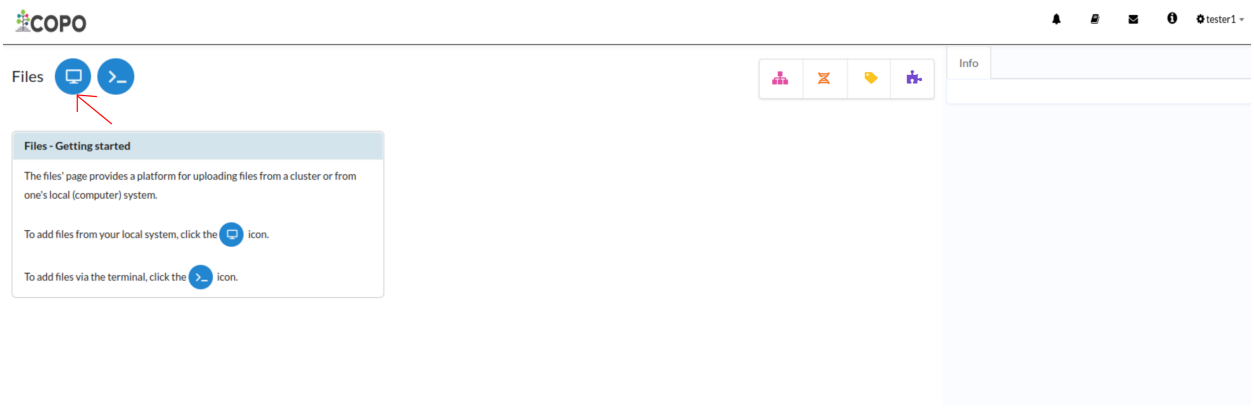


Fig. 3: **Files web page: 'Add new file by browsing local file system' button**

Fig. 4: **Files submission: Upload File dialogue**

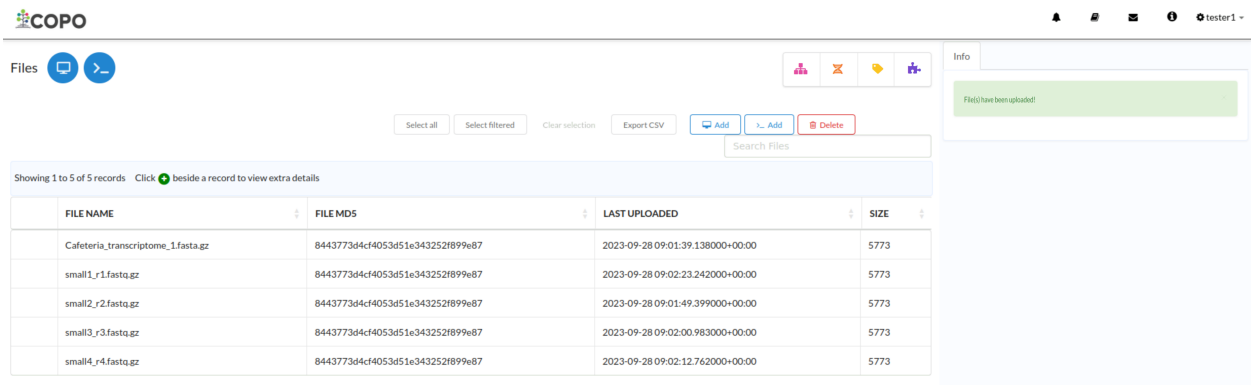


Fig. 5: **Files submission: Files' web page displaying the uploaded file(s)**

13.1.3 Submit Files via the Terminal

Hint: If you encounter the server certificate error, `curl: (60) Peer's Certificate issuer is not recognized.`, please perform one of the the following resolutions:

- **Resolution #1**: Run the command below in the terminal (if you have `sudo` rights on your device)

```
$ sudo apt-get install ca-certificates
```

OR

- **Resolution #2**: Replace `https` with `http` in the generated command indicated *here* then, run the command in the terminal again.

1. Click the button on the Files web page to add a new file from a cluster via the terminal.

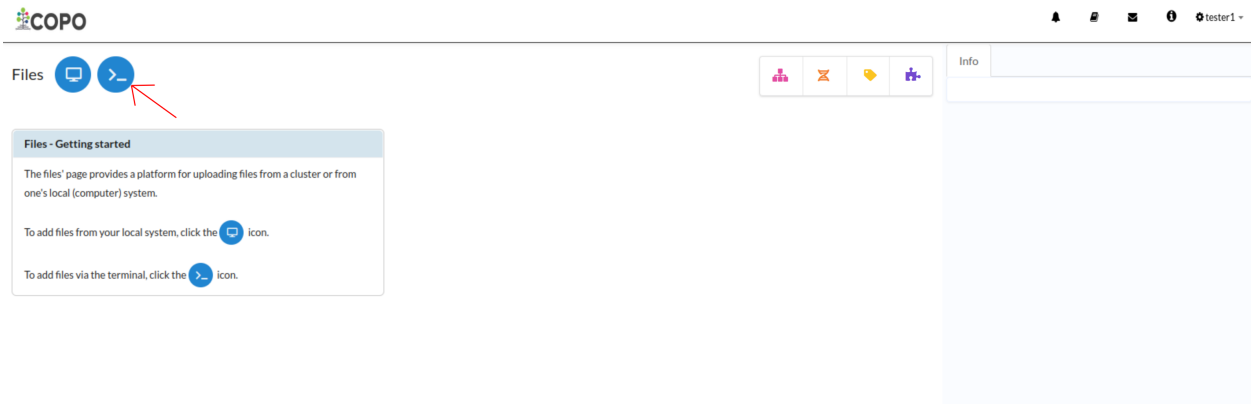


Fig. 6: **Files web page: 'Add new file via terminal' button**

2. A **Move Data** dialogue is displayed. Follow the instructions displayed then, click the **Process** button to submit the file(s) to the profile.

-
-
-
-

3. The new file(s) will be displayed on the **Files** web page after a successful file submission via the terminal i.e. after the command has been executed successfully in the terminal.

Hint: To add more files via the terminal, click the >_ Add button (once files have been submitted to the profile) as an alternative to clicking the button.

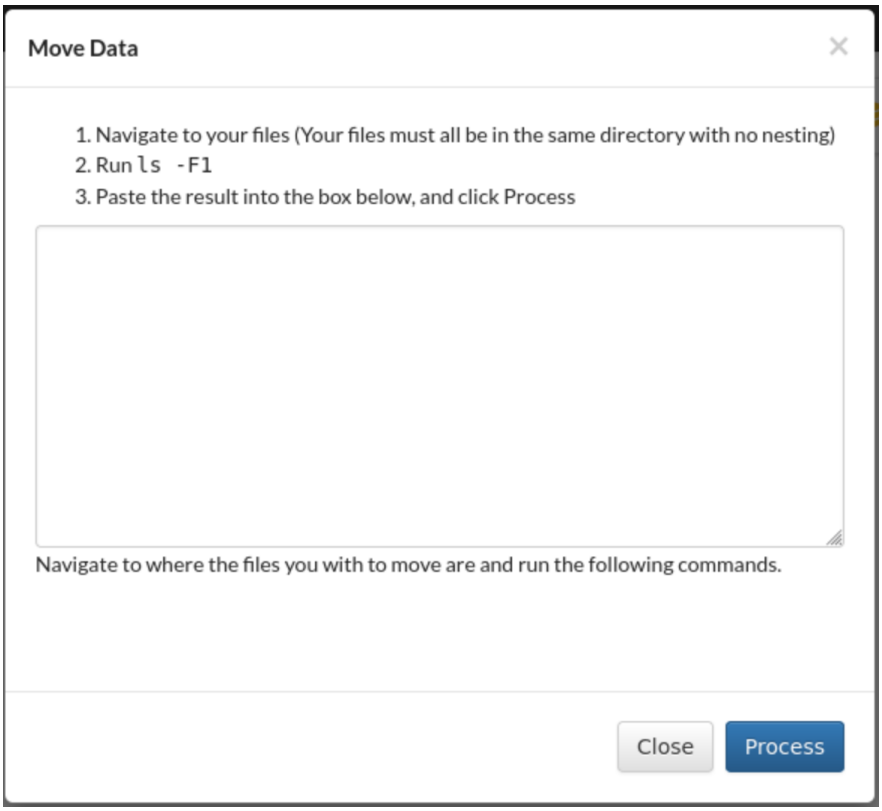


Fig. 7: **Files submission: Move Data dialogue**

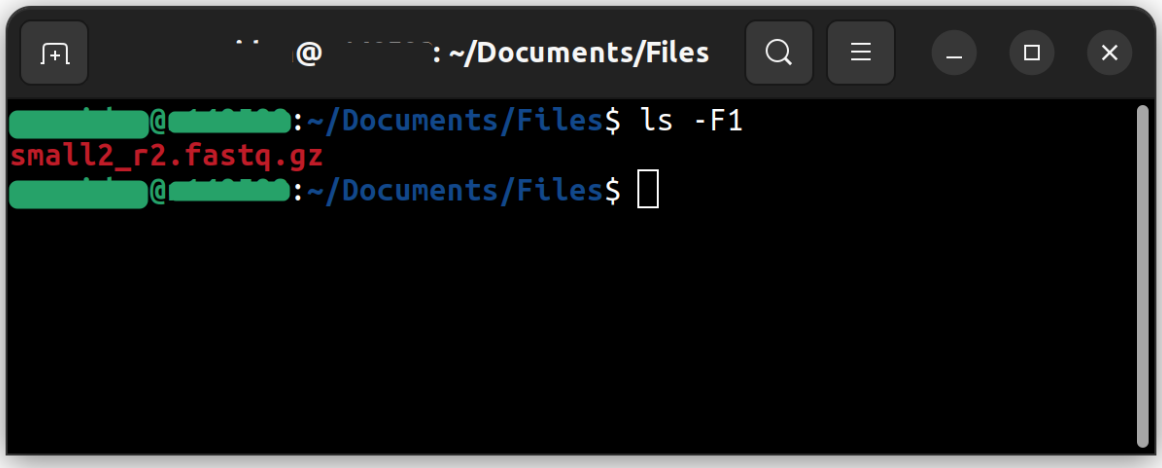


Fig. 8: **Input** `$ ls -F1` **command in the terminal**

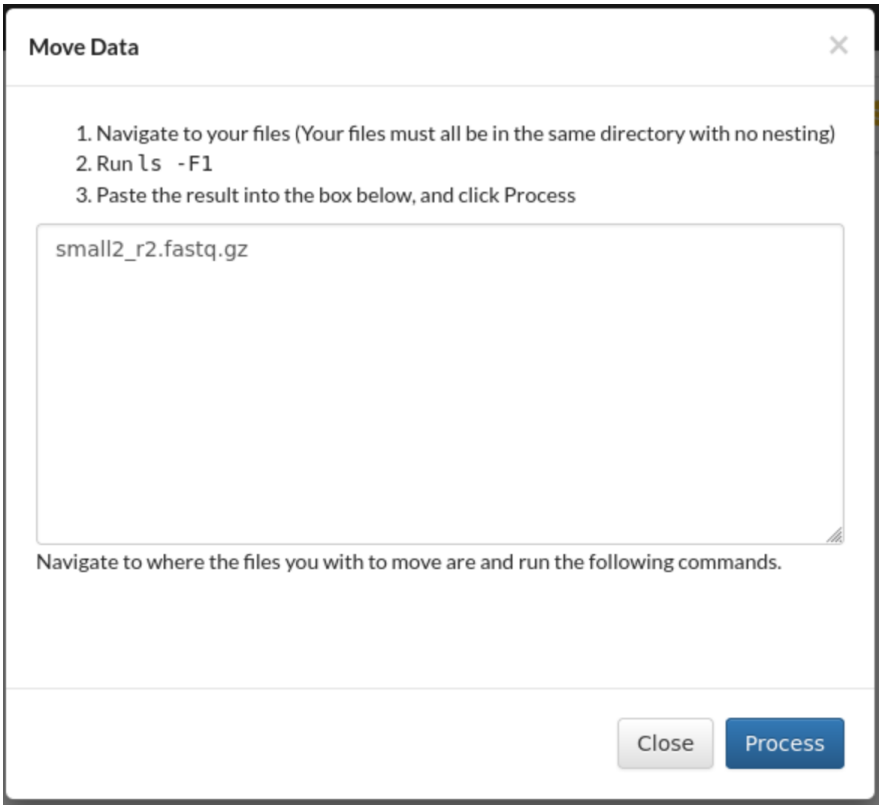


Fig. 9: **Move Data dialogue: Input the filename(s) returned after having ran the** `$ ls - F1` **command in the terminal. Then, click the** `Process` **button.**

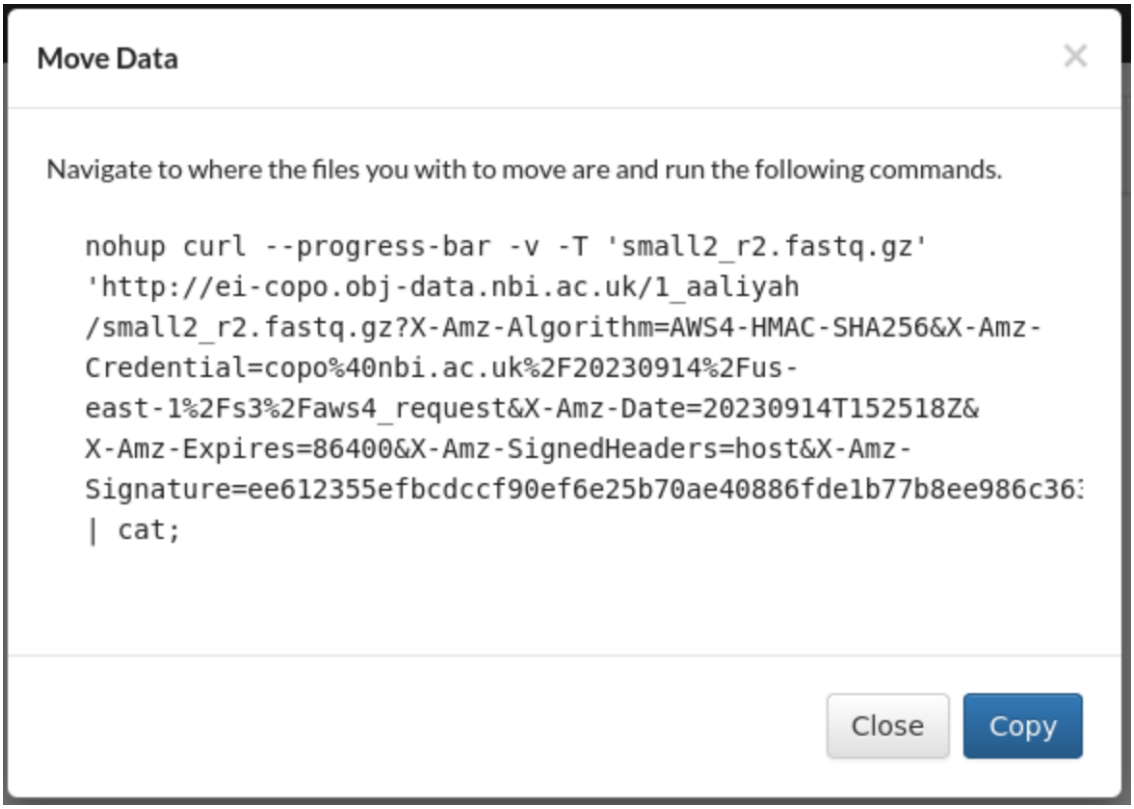


Fig. 10: **Move Data dialogue: Command outputted after having clicked command in the `Process` button. Copy the command displayed.**

If you encounter the server certificate error, `curl: (60) Peer's Certificate issuer is not recognized.`, please see the *hint* at the beginning of this section.

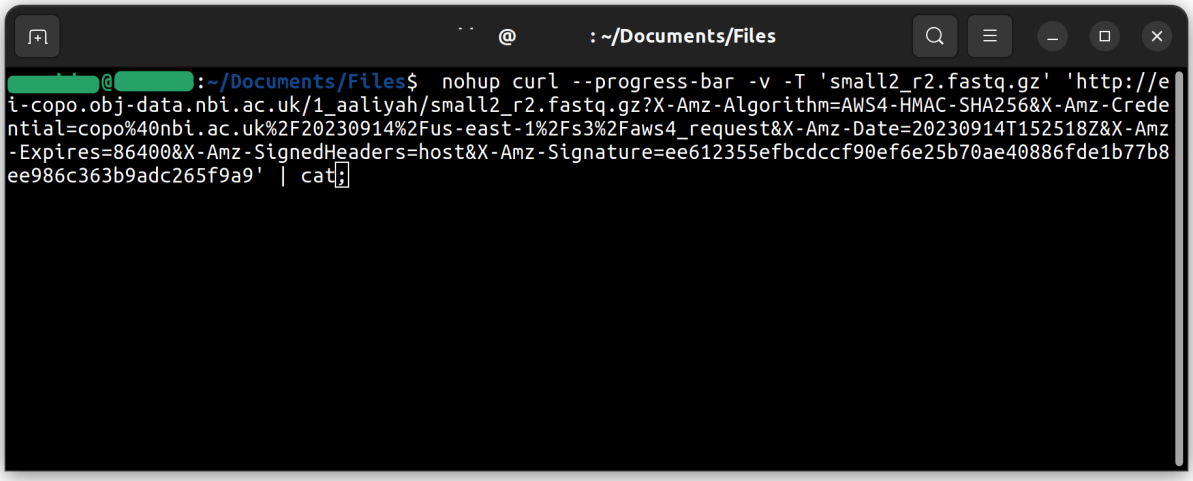


Fig. 11: **Paste the copied command in the terminal**

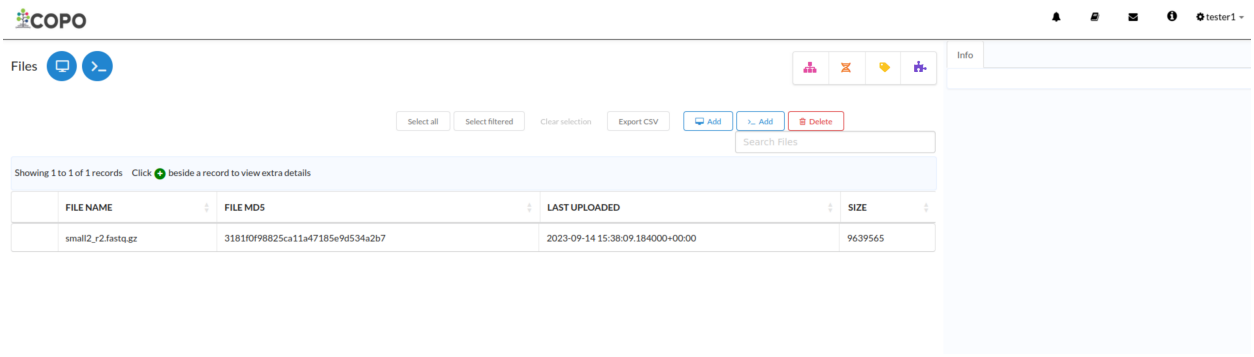


Fig. 12: **Files submission: Files' web page displaying the uploaded file(s)**

13.2 How to Delete Files

Click the desired file from the list of files displayed on the Files' web page. Then, click the **Delete** button (located in the top-right corner of the table) as shown below:

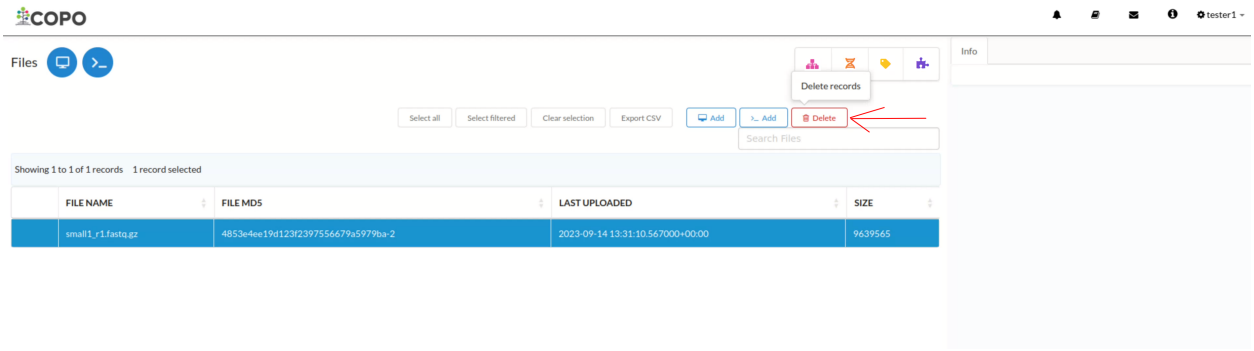


Fig. 13: **File deletion: Click the "Delete" button to remove the highlighted file from the profile**

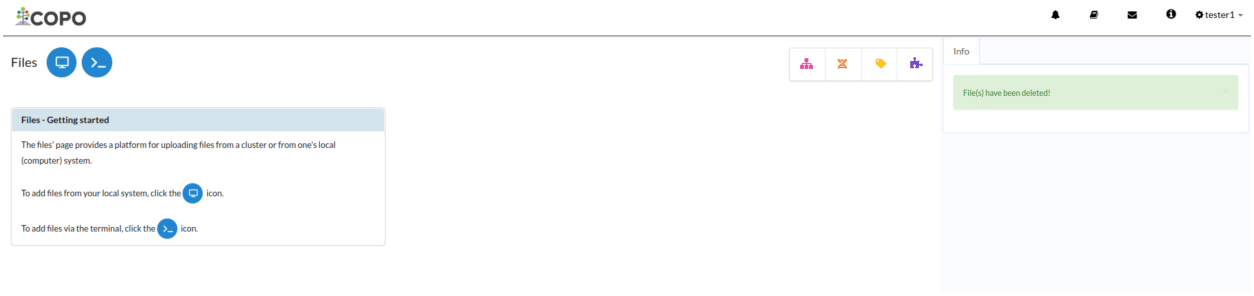


Fig. 14: **File deletion: File has been deleted**

CHAPTER

FOURTEEN

IMAGES SUBMISSION

14.1 How to Submit Images

Note:

- Images can only be submitted after samples have been uploaded in the **Upload Spreadsheet** dialogue.
- Images can only be submitted via a ToL[1] profile. Please see: *Steps to Create a Tree of Life Profile* for guidance.
- The filename of sample images must be named as `{Specimen_ID}-{n}.[jpg|png]` where `{n}` is the image number, `{Specimen_ID}` is the specimen ID of the sample in the manifest and `jpg` or `png` is the extension of the file.

See also:

- *Viewing Submitted Images*
- *Permits Submission*
- How to Submit ASG Manifests
- How to Submit DTOL Manifests
- How to Submit ERGA Manifests
- *How to Submit Barcoding Manifests*

1. The images table is empty under the **Sample Images** tab in the **Upload Spreadsheet** dialogue.

Fig. 1: **Manifest submission: 'Upload Spreadsheet' dialogue with no images uploaded**

[1] See term: *Tree of Life (ToL)*.

2. The images table is populated with the images uploaded under the **Sample Images** tab in the **Upload Spreadsheet** dialogue.

Fig. 2: **Manifest submission: 'Upload Spreadsheet' dialogue with image(s) uploaded**

3. Click the Finish button to submit the images and samples.

   A **Submit Samples** confirmation dialogue is displayed. If you decide to confirm the samples submission, click the **Confirm** button.

14.2 View Submitted Images

Note:

- Images can only be viewed **after** they have been submitted.
- Images **cannot** be deleted or modified after they have been submitted.
- The button will only appear on the web page if you are assigned as a sample manager.

Hint:

- To **select multiple** sample records, hold down the `Ctrl` key on your keyboard and click on the sample records that you would like to select.
- To **select a range** of sample records, hold down the `Shift` key on your keyboard and click on the first and last sample records that you would like to select.
- To **select all** sample records, click Select all
- To **select filtered** sample records, click Select filtered
- To **clear selection** of sample records, click Clear selection

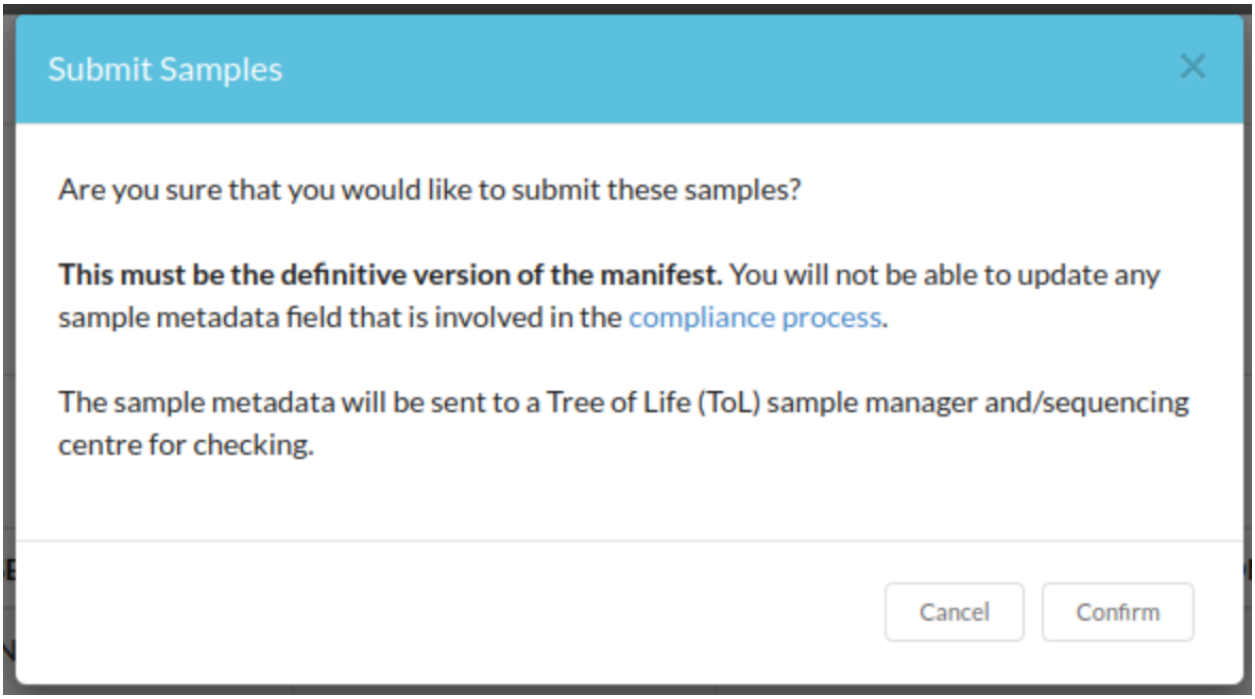


Fig. 3: **Manifest submission: 'Submit Samples' confirmation dialogue**

14.2.1 On Samples web page

1. Navigate to the **Samples** web page.

   See How to access Samples web page section for guidance.

2. Select the sample record(s) that you would like to view images for.

   Then, click the

button to view image(s) submitted for the selected sample record(s).

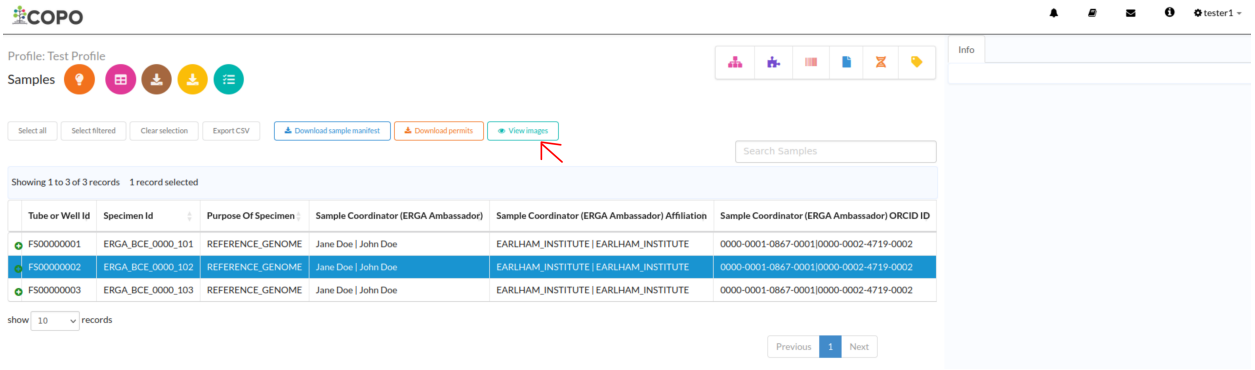


Fig. 4: **Samples web page: Pointer to 'View images' button**

3. If any image submission(s) exist for the selected sample record(s), a popup dialogue will be displayed with the image(s) submitted for the selected sample record(s) as shown below:

Hint: Click the image to view a larger version.

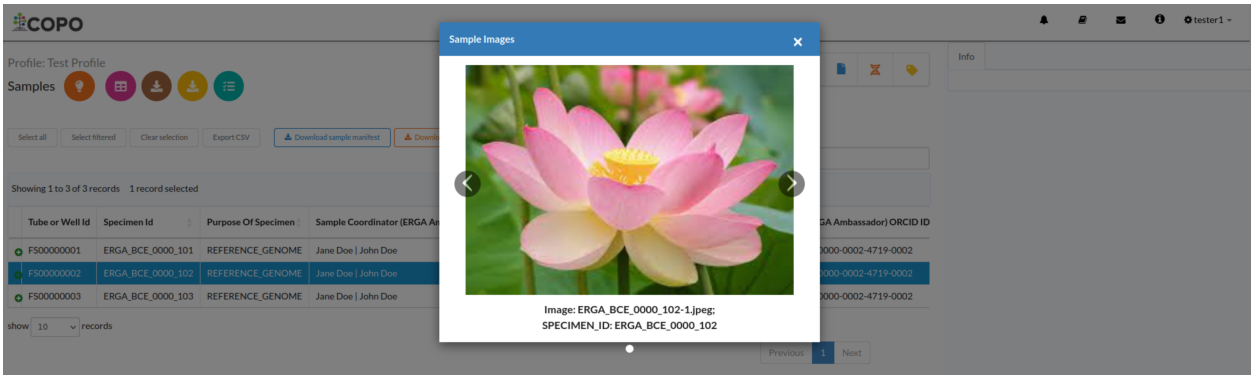


Fig. 5: **Samples web page: Popup dialogue displaying submitted image(s) for selected sample record(s)**

OR

If no images were submitted for the selected sample record(s), a message is displayed in the popup dialogue indicating such as shown below:

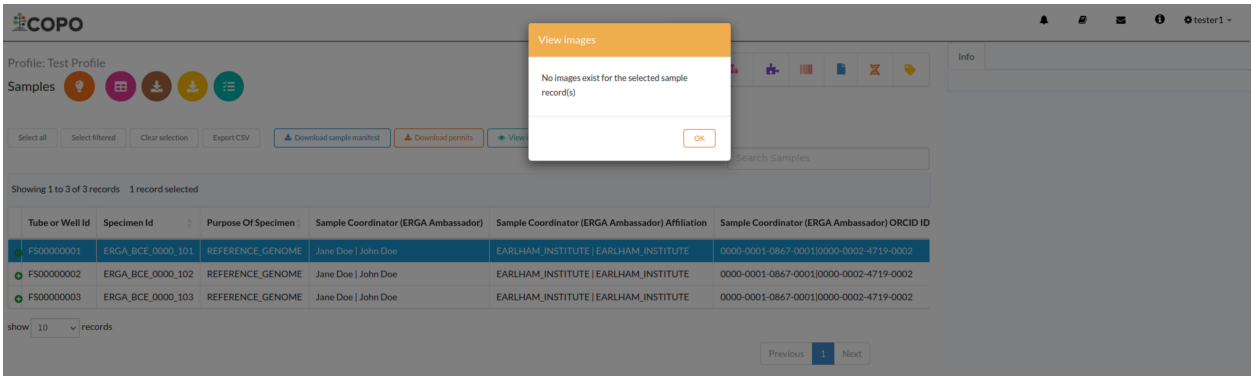


Fig. 6: **Samples web page: Popup dialogue displaying message, 'No images exist for selected sample record(s)'**

14.2.2 On Accept or Reject Samples web page

If you have been assigned as a **sample manager**, see *View submitted images section for sample managers <images-submission-view-images-sample-managers>* for more information.

CHAPTER

FIFTEEN

PERMITS SUBMISSION

15.1 How to Submit Permits

Note:

- Permits can only be submitted after ERGA[1] samples have been uploaded in the **Upload Spreadsheet** dialogue.
- Permits can only be submitted via a ToL[2] profile[3]. Please see: *Steps to Create a Tree of Life Profile* for guidance.

Warning:

- If you have more than one permit file to upload, they **must** be uploaded at the same time i.e. after you have clicked the Upload Permits button, navigate to the directory where the permits are stored and `CTRL + click` all of the permits so that all the permits are highlighted and uploaded at the same time.

See also:

- *Downloading Submitted Permits*
- *Images Submission*
- How to Submit ERGA Manifests
- *How to Submit Barcoding Manifests*

1. The uploaded samples are shown in a table in the **Upload Spreadsheet** dialogue as shown below. Click the Upload Permits button to browse your local (computer) system for `.pdf` permits for the samples.

2. The permits table is empty under the **Sample Permits** tab in the **Upload Spreadsheet** dialogue.

3. The permits table is populated with the permits uploaded under the **Sample Permits** tab in the **Upload Spreadsheet** dialogue

[1] See term: *ERGA*.
[2] See term: *Tree of Life (ToL)*.
[3] Also known as COPO profile. See: *COPO profile or work profile*.

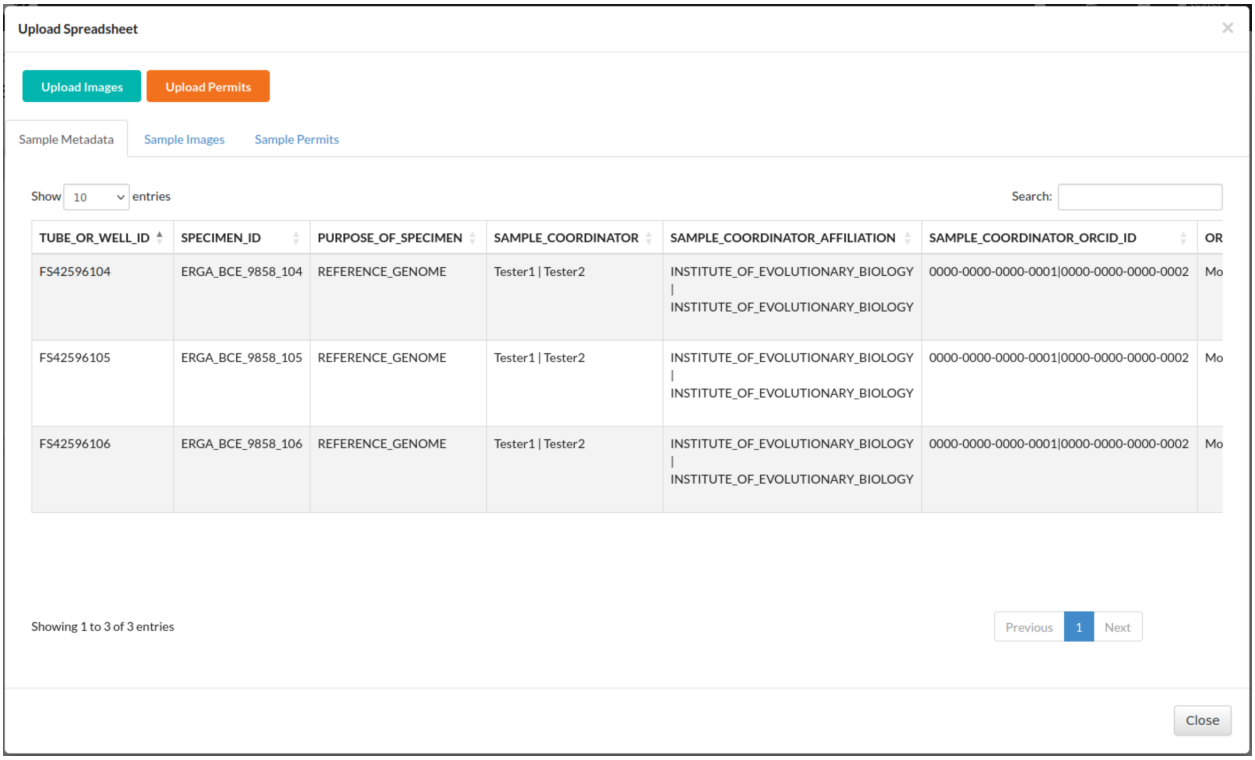


Fig. 1: **ERGA manifest submission: 'Upload Spreadsheet' dialogue with uploaded samples**

Fig. 2: **ERGA manifest submission: 'Upload Spreadsheet' dialogue with no permits uploaded**

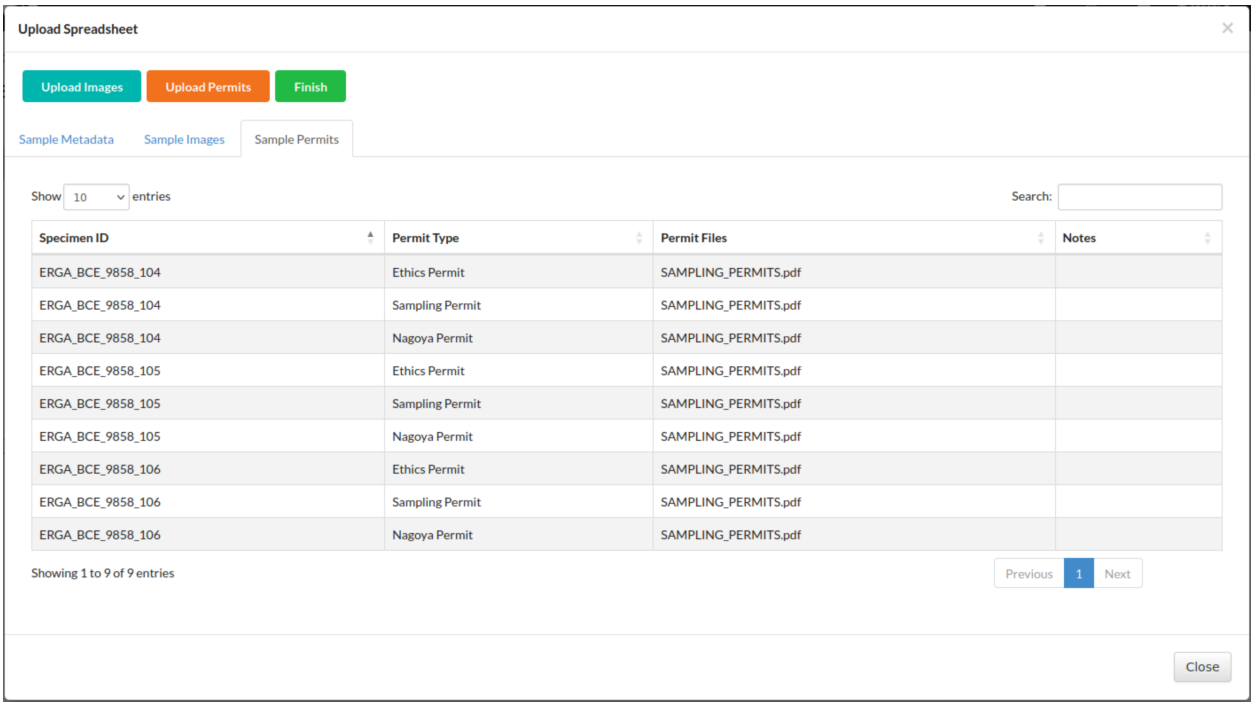


Fig. 3: **ERGA manifest submission: 'Upload Spreadsheet' dialogue with permits uploaded**

4. Click the Finish button to submit the permits and samples.

   A **Submit Samples** confirmation dialogue is displayed. If you decide to confirm the samples submission, click the **Confirm** button.

15.2 Download Submitted Permits

Note:

- Permits can only be downloaded **after** they have been submitted.
- Permits **cannot** be deleted or modified after they have been submitted.
- The button will only appear on the web page if you are assigned as a sample manager.

Hint:

- To **select multiple** sample records, hold down the `Ctrl` key on your keyboard and click on the sample records that you would like to select.
- To **select a range** of sample records, hold down the `Shift` key on your keyboard and click on the first and last sample records that you would like to select.
- To **select all** sample records, click Select all

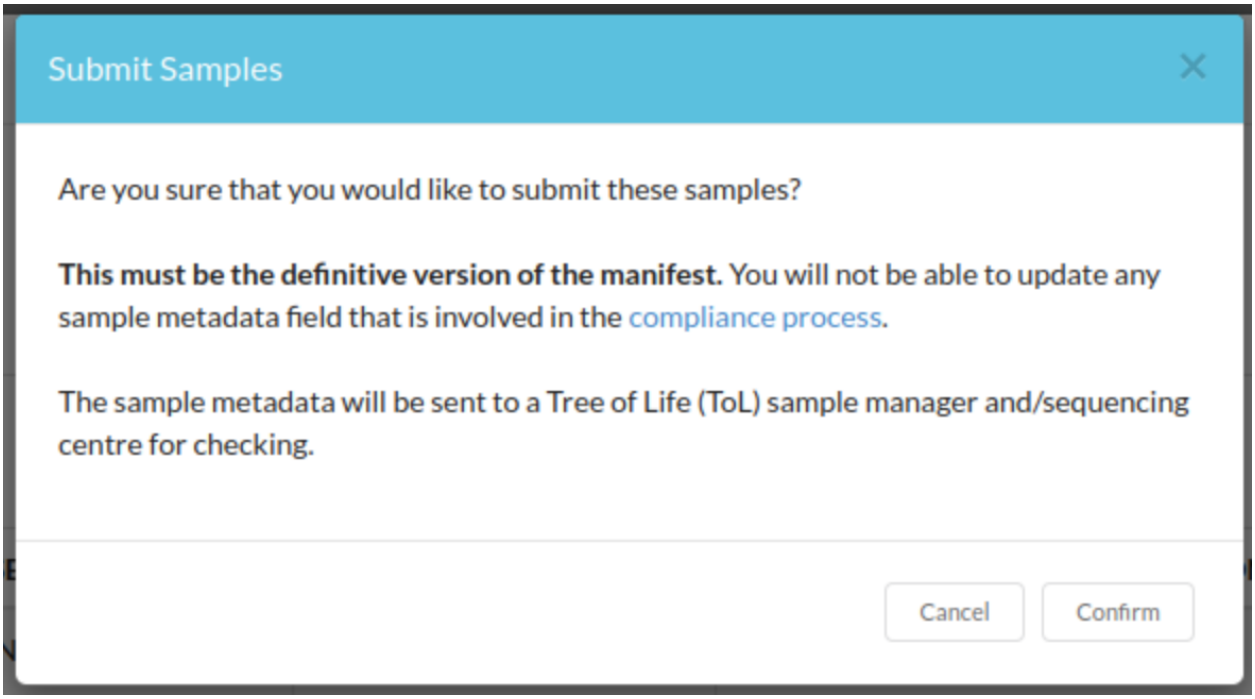


Fig. 4: **ERGA manifest submission: 'Submit Samples' confirmation dialogue**

- To **select filtered** sample records, click

- To **clear selection** of sample records, click

15.2.1 On Samples web page

1. Navigate to the **Samples** web page.

   See How to access Samples web page section for guidance.

2. Select the sample record(s) that you would like to download permits for.

   Then, click the

   button to download permit(s) submitted for the selected sample record(s).

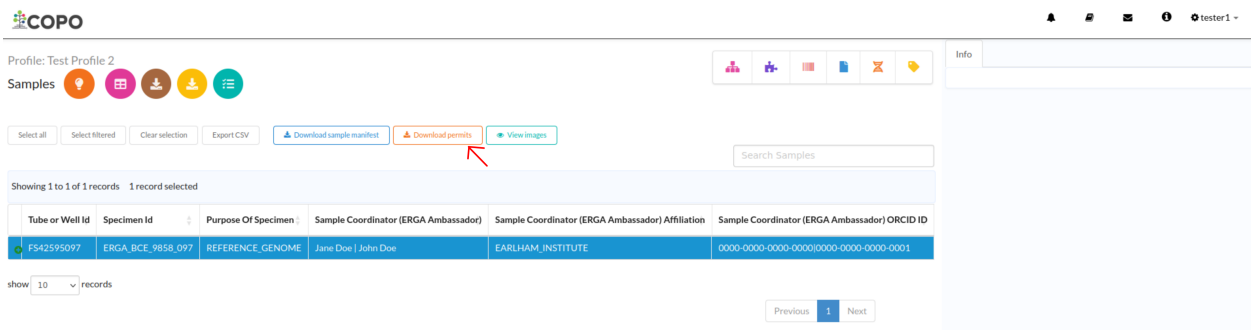


Fig. 5: **Samples web page: Pointer to 'Download permits' button**

3. If any permit submission(s) exist for the selected sample record(s), the permits will be automatically downloaded for the selected sample record(s) as shown below:

   Hint: Permits will be downloaded as a `.zip` file

   If no permits were submitted for the selected sample record(s), a message is displayed in the popup dialogue indicating such as shown below:

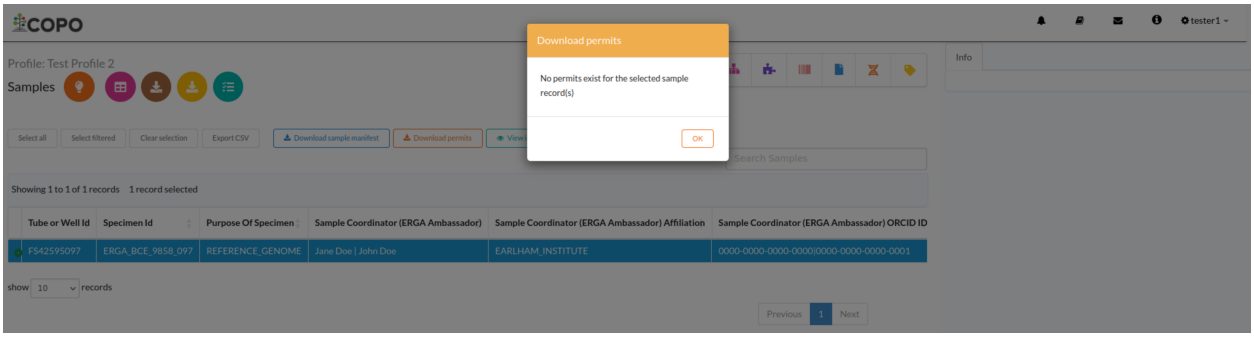


Fig. 6: **Samples web page: Popup dialogue displaying message, 'No permits exist for selected sample record(s)'**

15.2.2 On Accept or Reject Samples web page

If you have been assigned as a **sample manager**, see *Download submitted permits section for sample managers <permits-submission-download-permits-sample-managers>* for more information.

CHAPTER

SIXTEEN

READS SUBMISSION

16.1 How to Submit Reads

Note:

- Once **Reads** have been submitted, they cannot be deleted.
- Files must be uploaded before a **Reads** manifest can be submitted.

See also:

- *How to Update Reads*
- *Explore Various Types of Reads Checklist*
- *How to Submit Files*

16.1.1 Accessing the Reads' Web Page

The **Reads'**[1] web page can be accessed from the **Components** button or **Actions** button associated with a profile[2].

Use Components' Button to Navigate to Reads' Web Page

Click the

button associated with a profile. Then, click the

from the popup menu displayed as shown below:

Use Actions' Button to Navigate to Reads' Web Page

Click the

button associated with a profile. Then, click the action, **Submit Reads** action from the popup menu displayed as shown below:

[1] See: *Reads*.
[2] Also known as COPO profile. See: *COPO profile or work profile*.

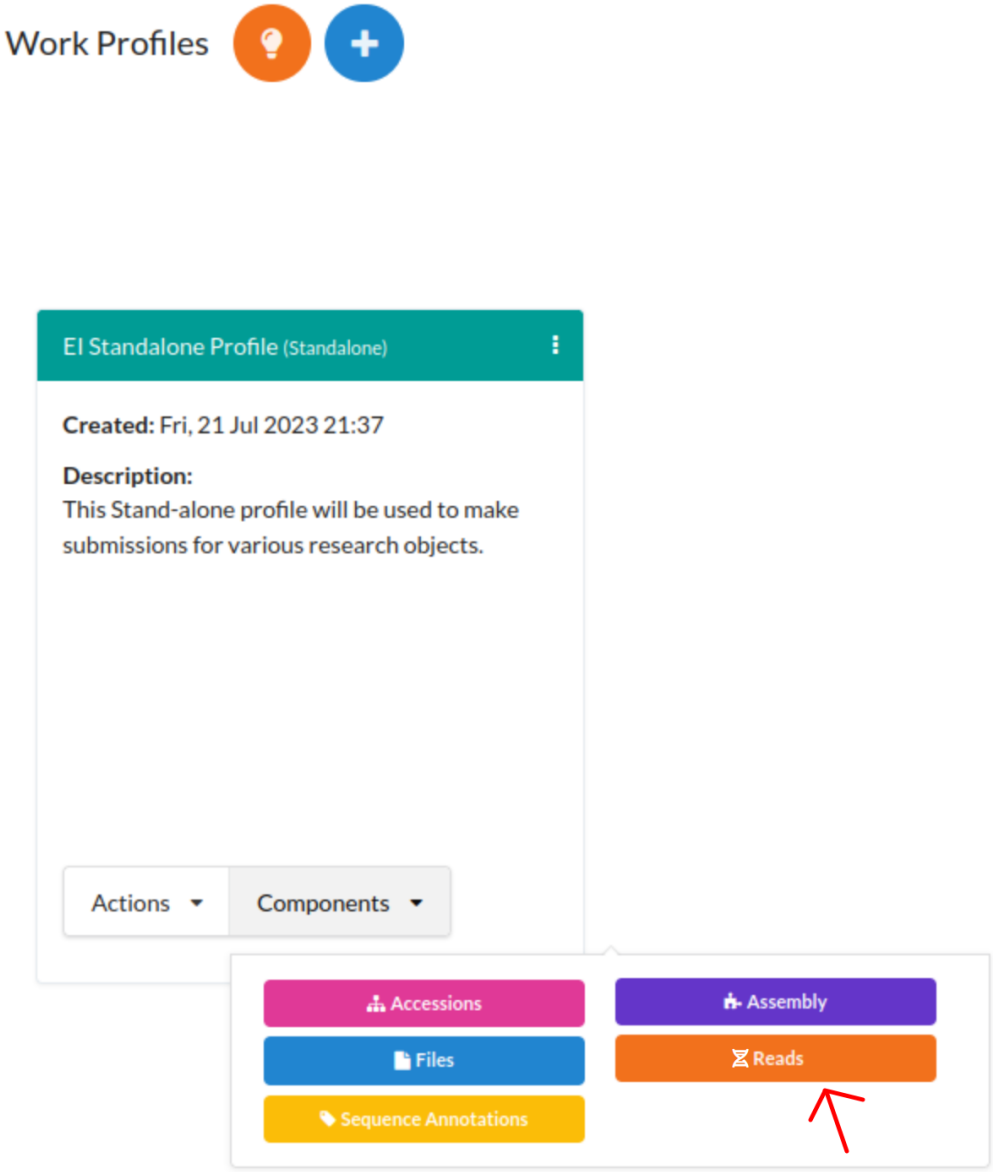


Fig. 1: **Stand-alone Profile Components: Reads component button**

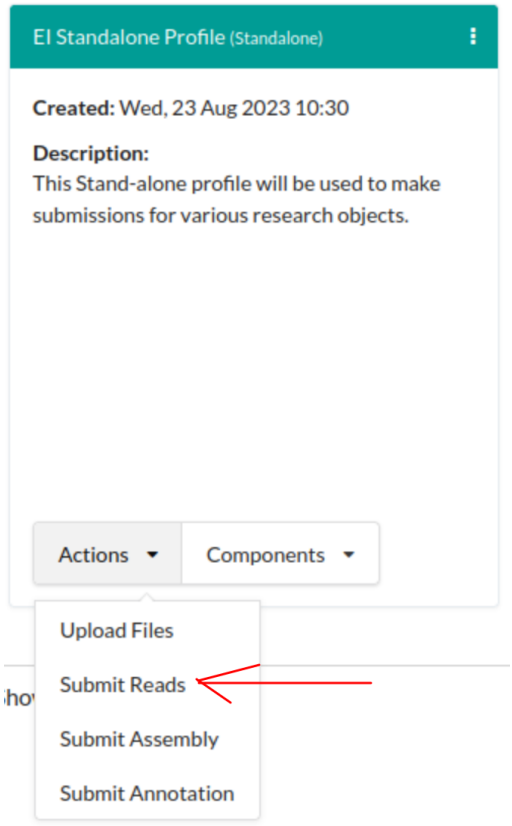


Fig. 2: **Stand-alone Profile Actions: 'Submit Reads' action**

16.1.2 Upload Reads

Hint: To download a blank **Reads'** manifest template, click the button.

Important: The **Reads** checklist dropdown menu is only available for **Standalone** profiles.

If you are submitting **Reads** for a profile other than a Standalone profile, please proceed to upload the Reads from your local file system after you have clicked the button.

1. On the **Reads'** web page, click the checklist dropdown to view a list of available checklists that support **Reads'** submission as shown below:

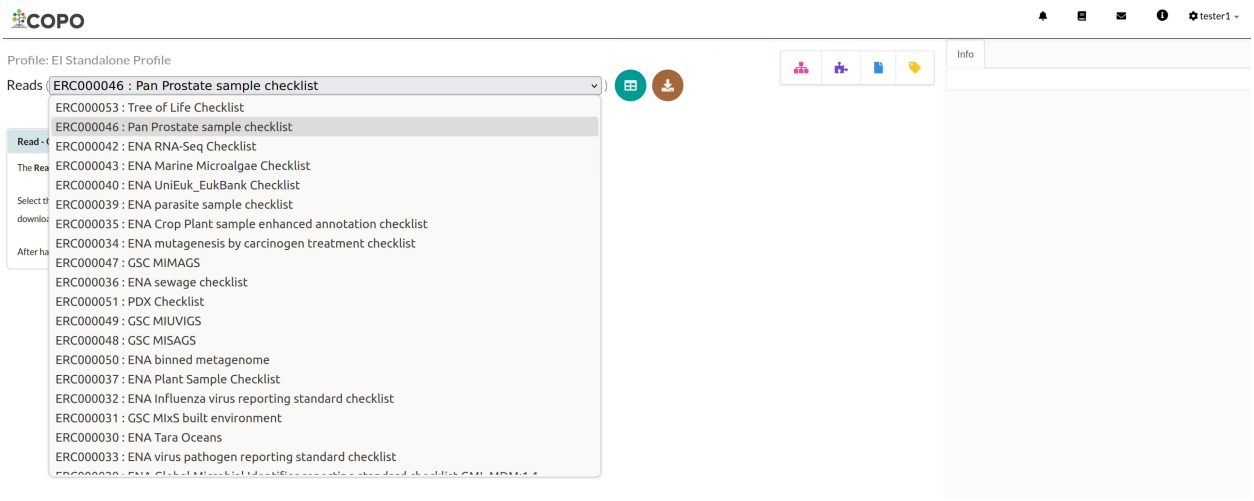


Fig. 3: **Reads' web page: Checklist dropdown menu with checklist options displayed**

An overview of each **Reads** checklist option is explained in the *Reads' checklist section*.

2. Click button to add **Reads'** from a spreadsheet for the chosen checklist as shown below:

> **Note:** The colour of the button is based on the type of profile that you are submitting a Reads for. See the *Profile Types Legend* section regarding the colour code for the various types of project profiles on COPO.

3. An **Upload Read Manifest** dialogue is displayed. Click the Upload Read Manifest button to choose a file from your local system.

4. The uploaded manifest is shown in a table in the **Upload Read Manifest** dialogue as shown below. Click the Finish button to submit the reads manifest.

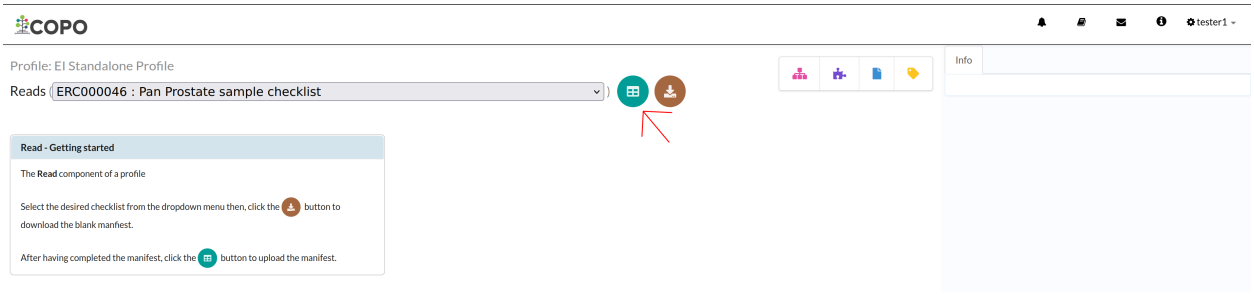


Fig. 4: **Reads' upload: Click 'Add Reads' from Spreadsheet' button**

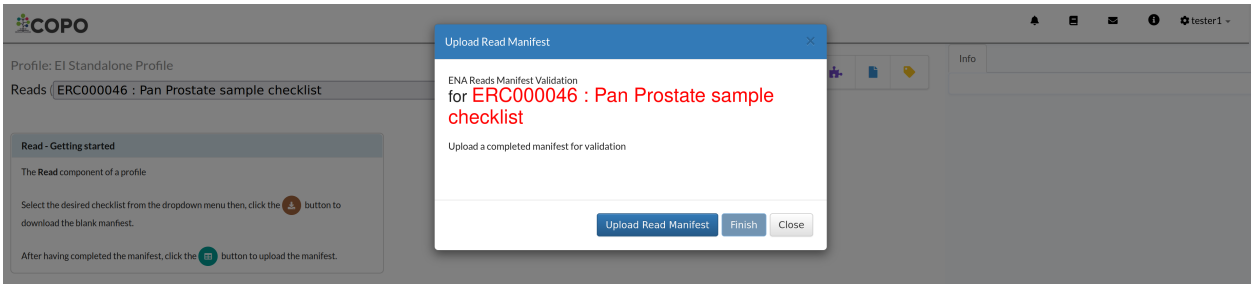


Fig. 5: **Reads' upload: 'Upload Read Manifest' dialogue**

5. The new read(s) will be displayed on the **Reads** web page after a successful submission.

> **Hint:** Reads records that are highlighted **yellow** indicate that the records are pending submission. The records will be highlighted **green** after a successful submission.

16.1.3 Submit Reads

Hint: The submitted read record will be highlighted **green**.

Click the desired reads records from the list of reads displayed on the **Reads'** web page. Then, click the **Submit** button (located in the top-right corner of the table) as shown below:

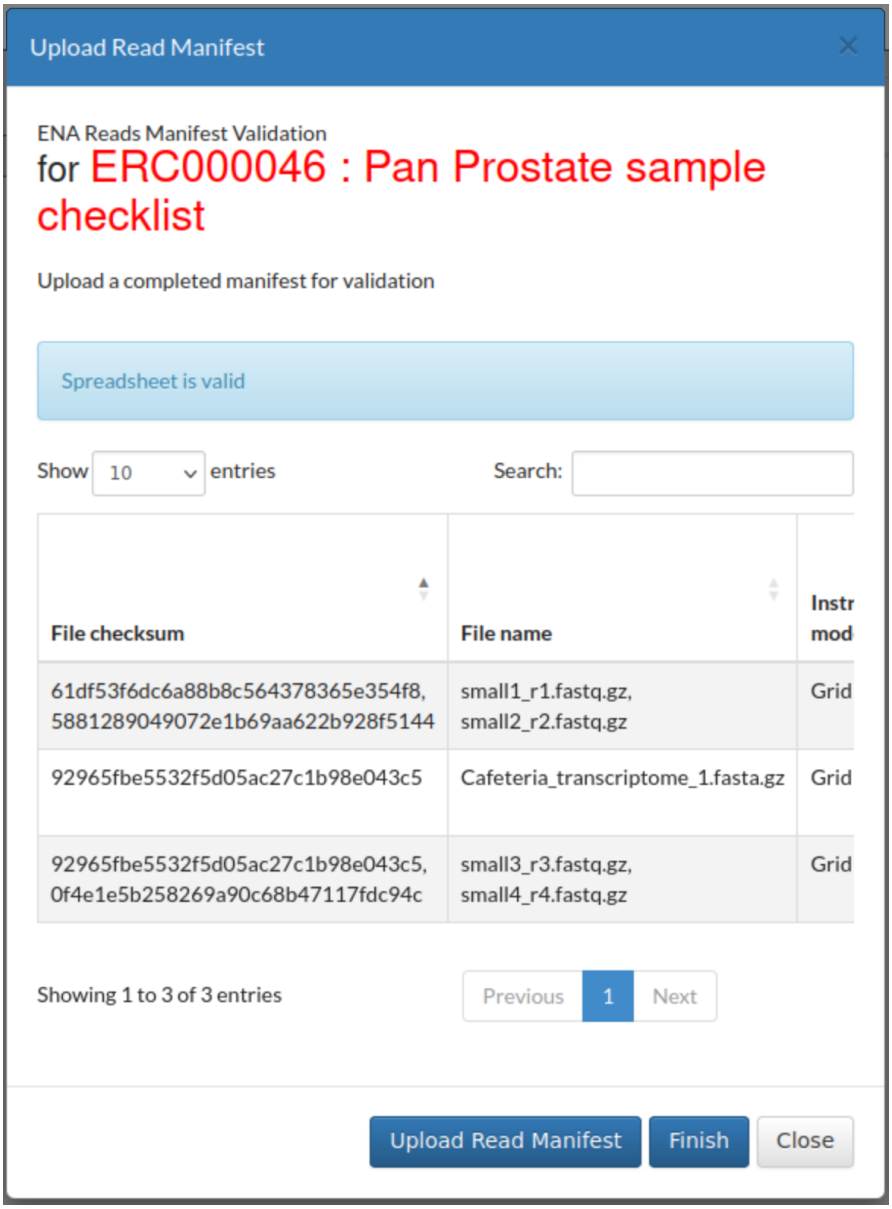


Fig. 6: **Reads' upload: 'Upload Read Manifest' dialogue with uploaded manifest**

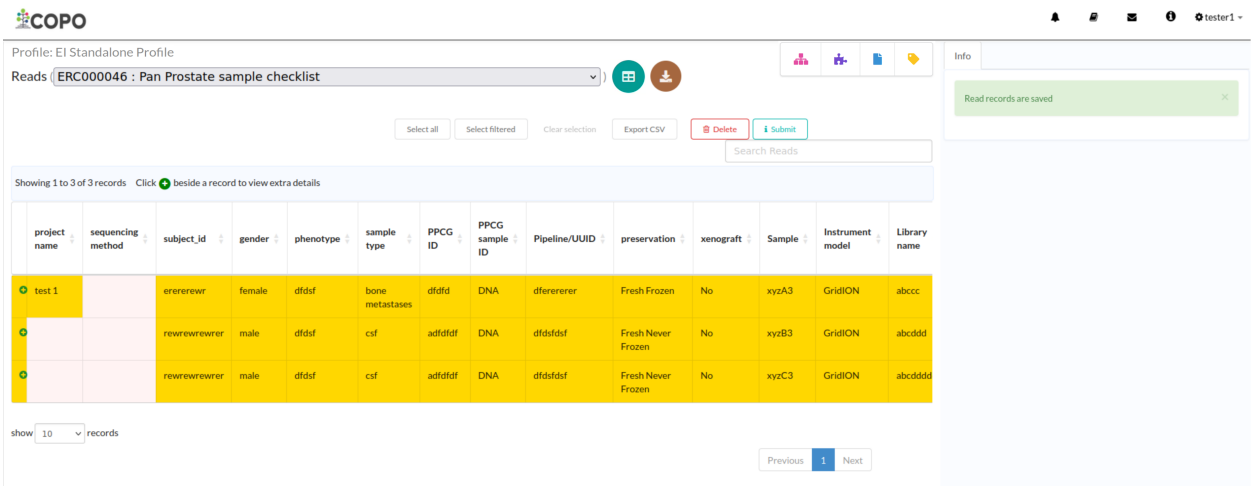


Fig. 7: **Reads' upload: Reads' web page displaying the uploaded read(s)**

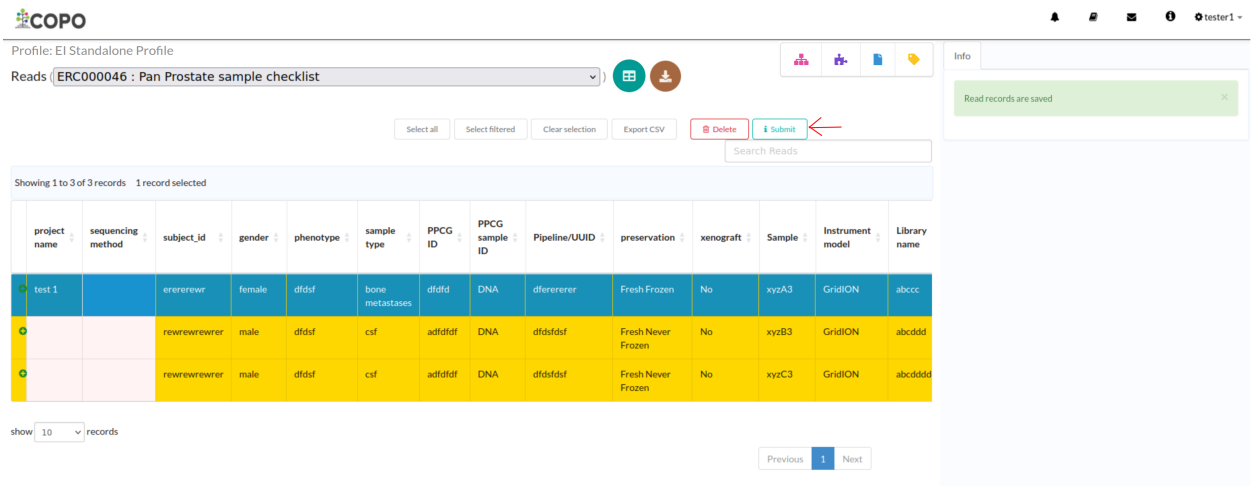


Fig. 8: **Reads submission: Click the "Submit" button to submit the highlighted read from the profile**

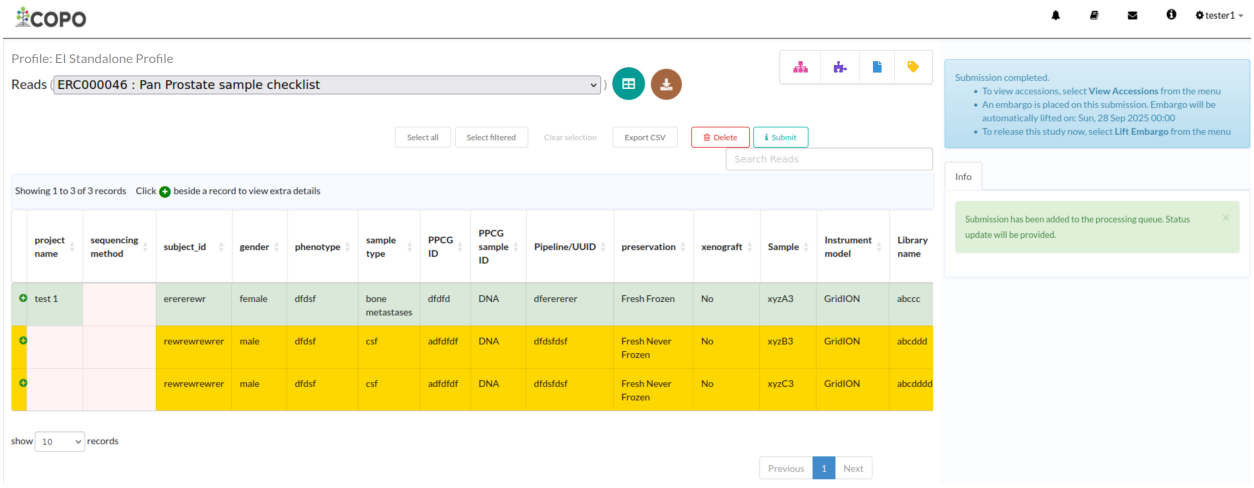


Fig. 9: **Reads submission: The Read has been submitted**

CHAPTER

SEVENTEEN

SAMPLES SUBMISSION

The following are the types of sample manifests that can be submitted.

Click the links below to learn more about each type of manifest submission:

- Aquatic Symbiosis Genomics (ASG) manifests
- Darwin Tree of Life (DToL) manifests
- Darwin Tree of Life Environmental (DToL_ENV) manifests
- European Reference Genome Atlas (ERGA) manifests

See also:

- How to access Samples web page
- *How to submit Assemblies*
- *Barcoding manifest submissions*
- *How to upload Files*
- *How to submit Reads*
- *How to submit Sequence Annotations*

17.1 How to Download Submitted Sample Manifest

Hint: This is useful if you would like to update sample metadata for a manifest or retrieve the actual manifest that was submitted.

Samples can be updated by resubmitting the manifest with the updated metadata. See *Updating Samples* section for more information about which fields can be updated.

Note: At least one sample record (in a manifest) must be submitted before a manifest can be downloaded.

The following steps describe how to download a submitted sample manifest:

1. Navigate to the **Samples** web page.

   See How to access Samples web page section for guidance.

2. On the **Samples** web page, select **only one** sample record from the sample record table displayed.

   Then, click the

button to download the manifest.

   Hint: The manifest will be automatically downloaded as a `.xlsx` file

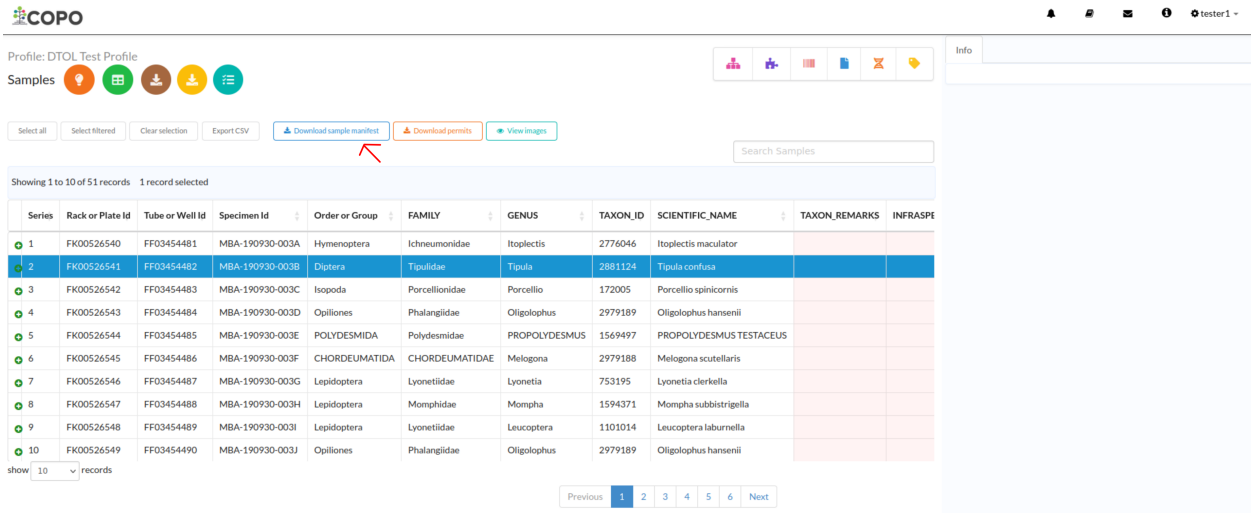


Fig. 1: **Samples web page: Pointer to 'Download sample manifest' button**

CHAPTER

EIGHTEEN

SEQUENCE ANNOTATIONS SUBMISSION

Note:

- Reads must be uploaded before a **Sequence Annotation** can be submitted.

See also:

- *How to Update Sequence Annotations*
- *How to Delete Sequence Annotations*

18.1 How to Submit Sequence Annotations

18.1.1 Accessing the Sequence Annotations' Web Page

The **Sequence Annotations'**[1] web page can be accessed from the **Components** button or **Actions** button associated with a profile[2].

Use Components' Button to Navigate to Sequence Annotations' Web Page

Click the

button associated with a profile. Then, click the

from the popup menu displayed as shown below:

Use Actions' Button to Navigate to Sequence Annotations' Web Page

Click the

button associated with a profile. Then, click the action, **Submit Annotation** action from the popup menu displayed as shown below:

[1] See: *Sequence Annotation*.
[2] Also known as COPO profile. See: *COPO profile or work profile*.

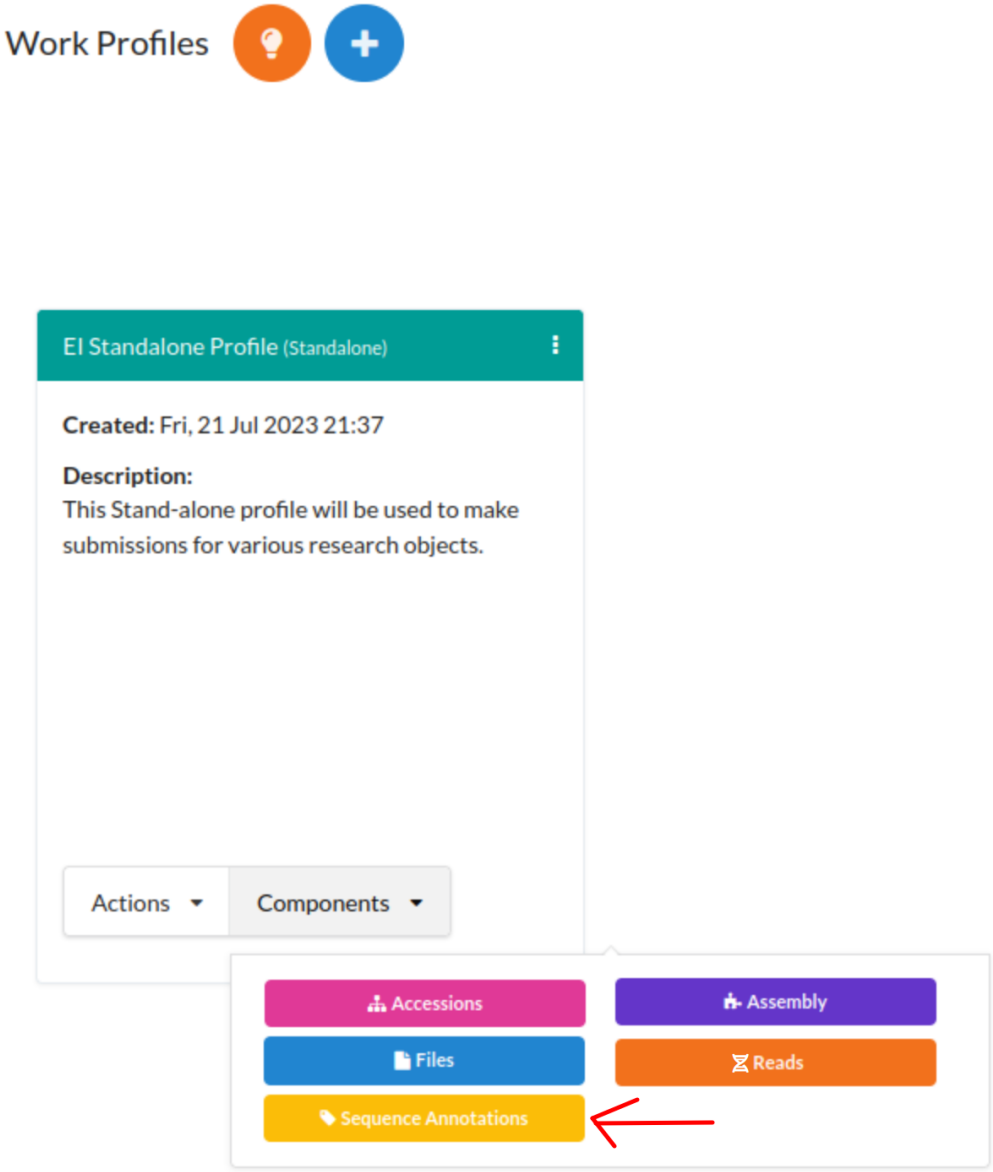


Fig. 1: **Stand-alone Profile Components: Sequence Annotations' component button**

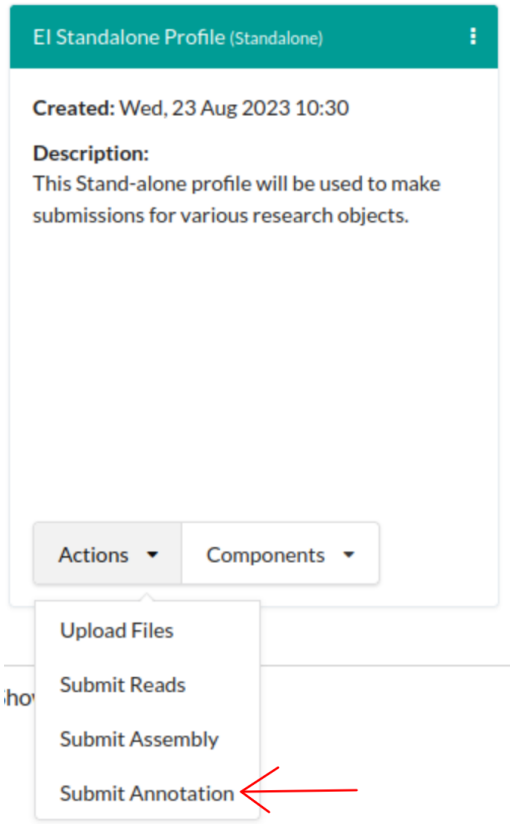


Fig. 2: **Stand-alone Profile Actions: 'Submit Annotation' action**

18.1.2 Submit Sequence Annotations

1. Click button to add **Sequence Annotations'** as shown below:

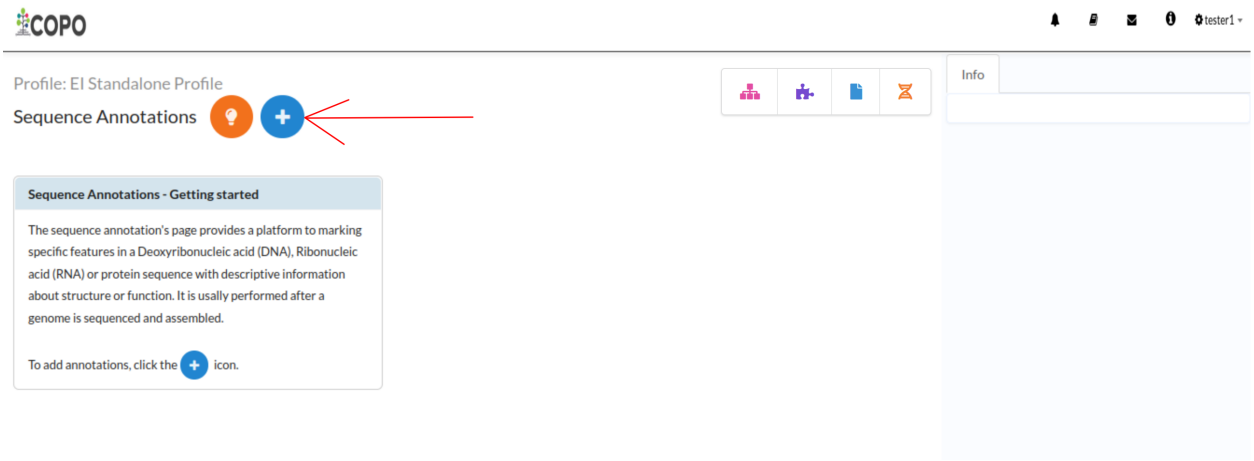


Fig. 3: **Sequence Annotations' submission: Click 'Add Record' button**

2. An **Add Sequence Annotation** dialogue is displayed. Provide the details then, click the **Submit Annotation** button.

   Select a sample accession from the **SAMPLE** dropdown field menu. The dropdown menu will display the reads that were previously uploaded.

3. The new sequence annotation(s) will be displayed on the **Sequence Annotations** web page after a successful submission.

   Hint: Sequence annotation records that are highlighted **yellow** indicate that the records are pending submission. The records will be highlighted **green** after a successful automatic submission.

   Sequence Annotations' submission: Sequence Annotations' web page displaying the submitted sequence annotation(s)

18.2 How to Delete Sequence Annotations

Note: Sequence annotations can only be deleted **before** they have been submitted.

Click the desired sequence annotation from the list of sequence annotations displayed on the **Sequence Annotations** web page. Then, click the **Delete** button (located in the top-right corner of the table) as shown below:

Fig. 4: **Sequence Annotations' submission: 'Submit Annotation' dialogue with no sample accession chosen from SAMPLE dropdown menu**

Fig. 5: **Sequence Annotations' submission: 'Submit Annotation' dialogue with sample accession chosen from SAMPLE dropdown menu**

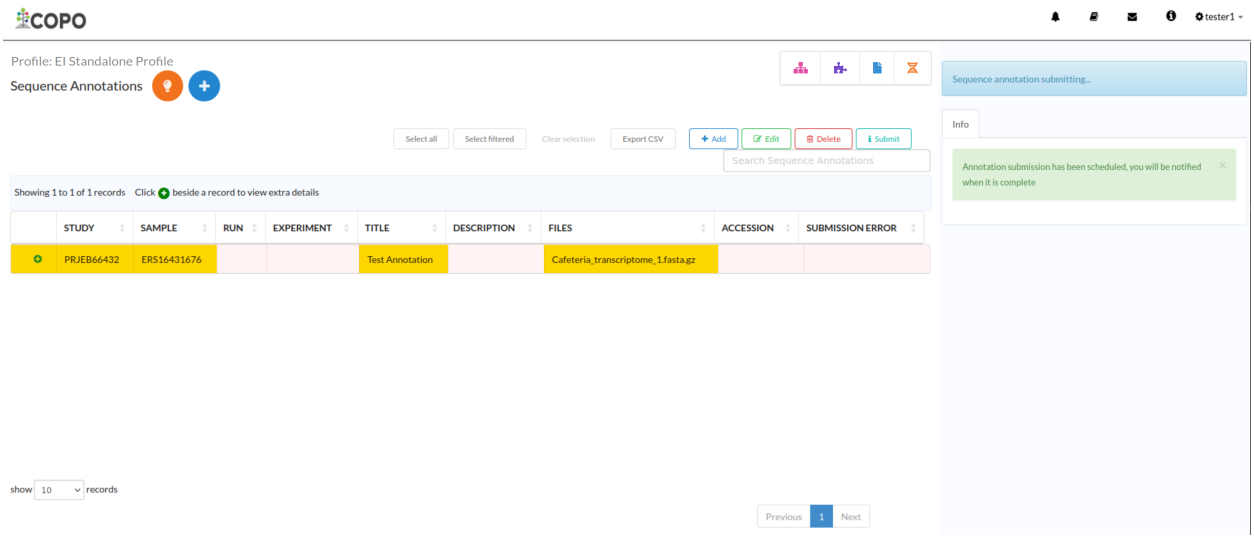


Fig. 6: **Sequence Annotations' submission: Sequence Annotations' web page displaying the (pending) uploaded sequence annotation(s)**

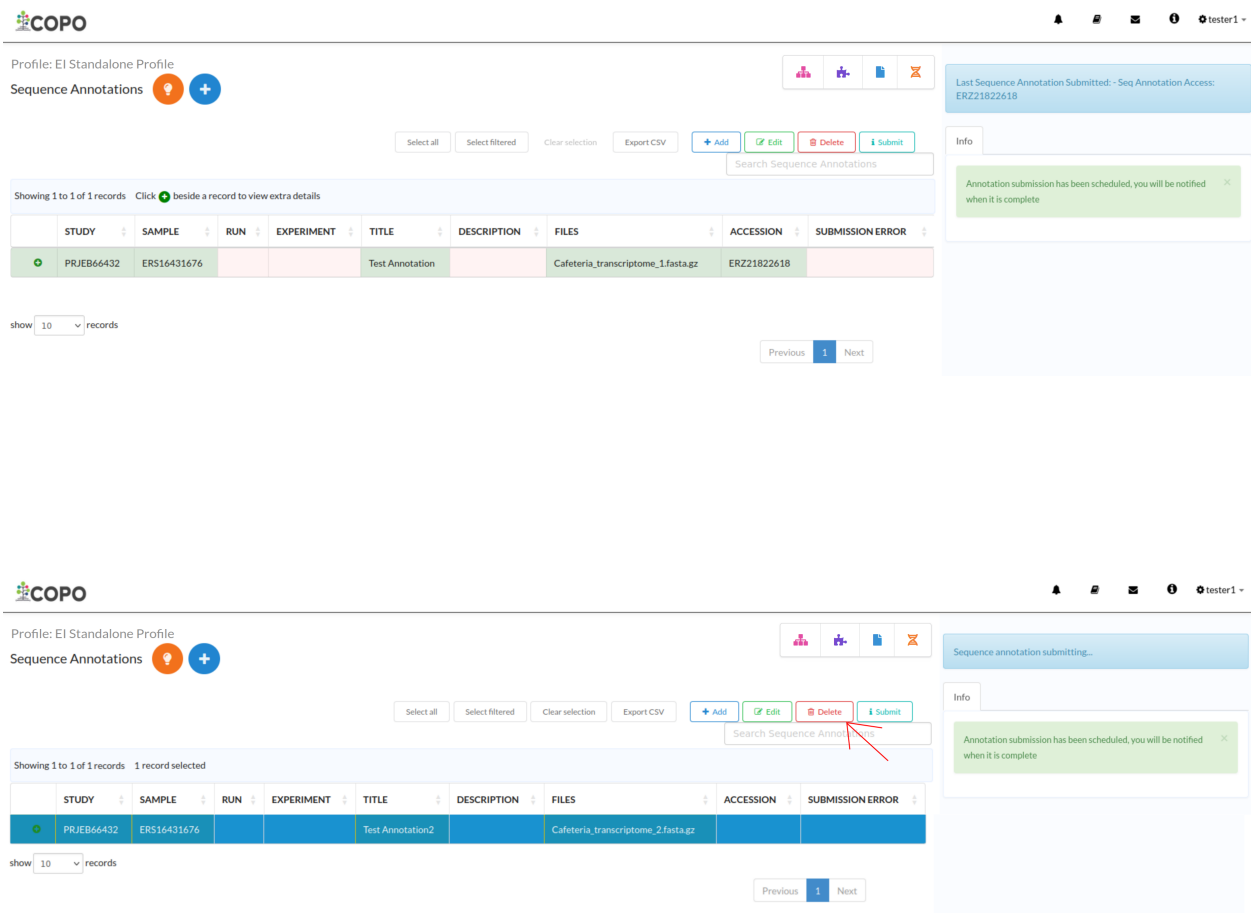


Fig. 7: **Sequence annotation deletion: Click the "Delete" button to remove the highlighted sequence annotation from the profile**

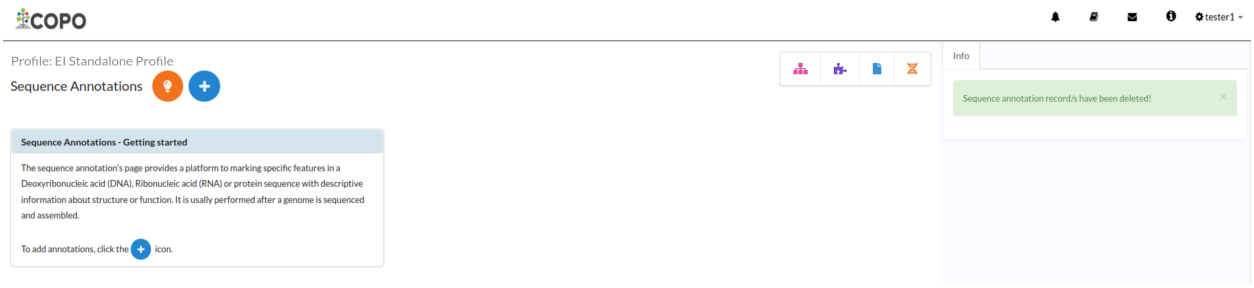


Fig. 8: **Sequence annotation deletion: Sequence annotation record has been deleted**

CHAPTER

NINETEEN

UPDATING PROFILES

Warning: Profile types cannot be changed or updated once they have been created. If you need to change the profile type, you will need to delete the profile and create a new one.

See also:

Deleting Profiles

To edit a profile, click the ⋮ icon that is associated with the desired profile. The option to edit or delete a profile record will be displayed once clicked.

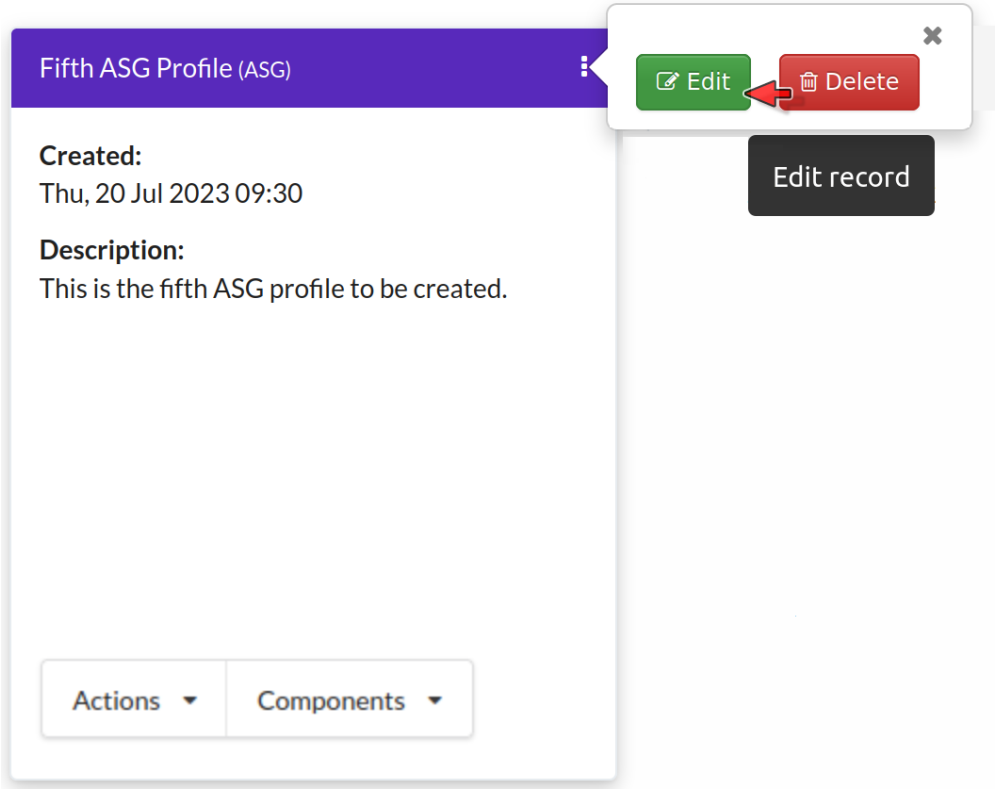


Fig. 1: **COPO Profile: Click the `Edit` button**

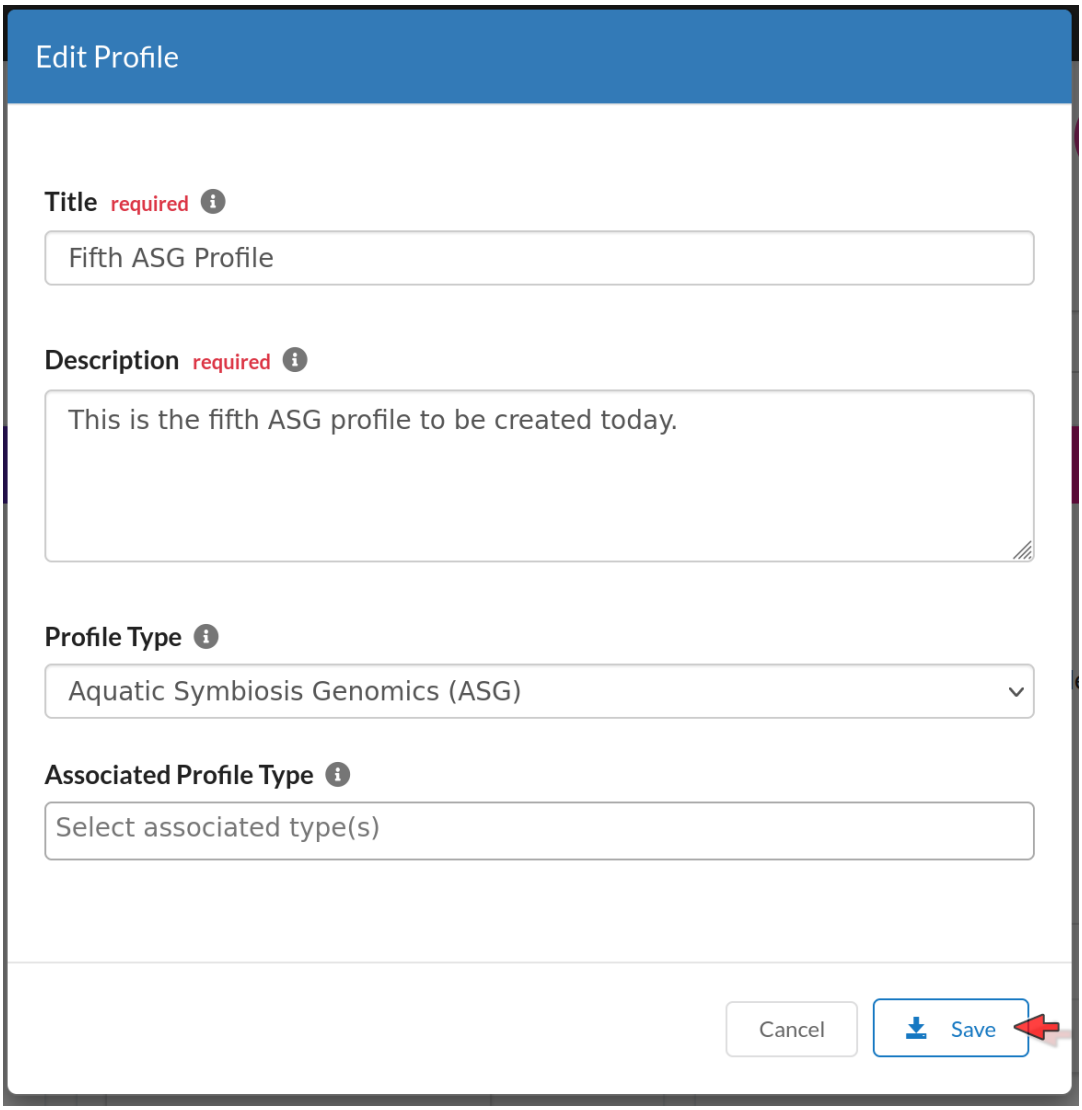


Fig. 2: **COPO Profile: Edit the existing details. Click the** Save **button to update the profile.**

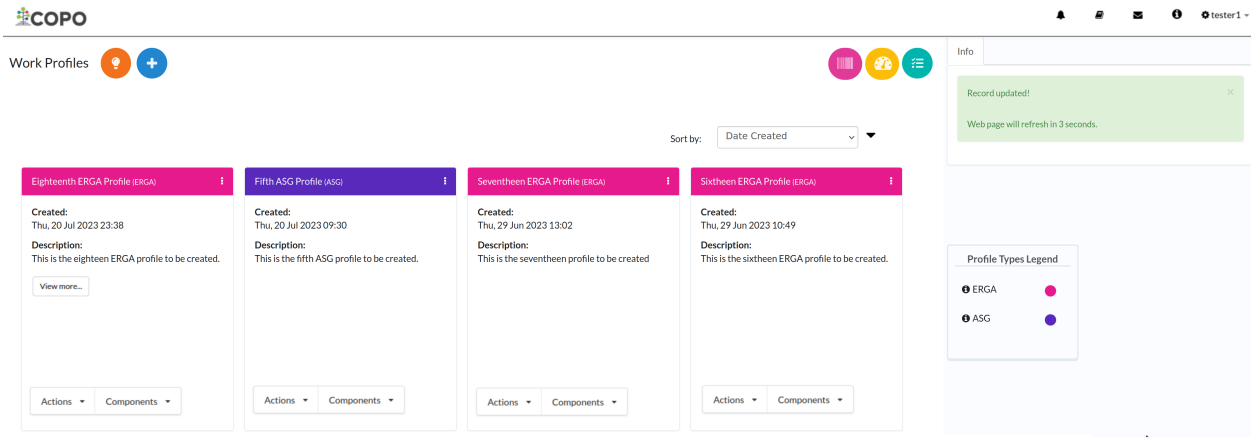


Fig. 3: **COPO Profile: Profile record has been updated. Wait 3 seconds for the web page to be automatically reloaded.**

19.1 Deleting Profiles

Note:

- Deleting a profile will **not** delete the associated COPO user account. The user account will still be able to log into the system.
- Only profiles that have no associated research objects such as samples, reads, assemblies or files etc. can be deleted.
- If you need to have a profile deleted that has associated research objects, please contact the .

To delete a profile, click the ⋮ icon that is associated with the desired profile. The option to edit or delete a profile record will be displayed once clicked.

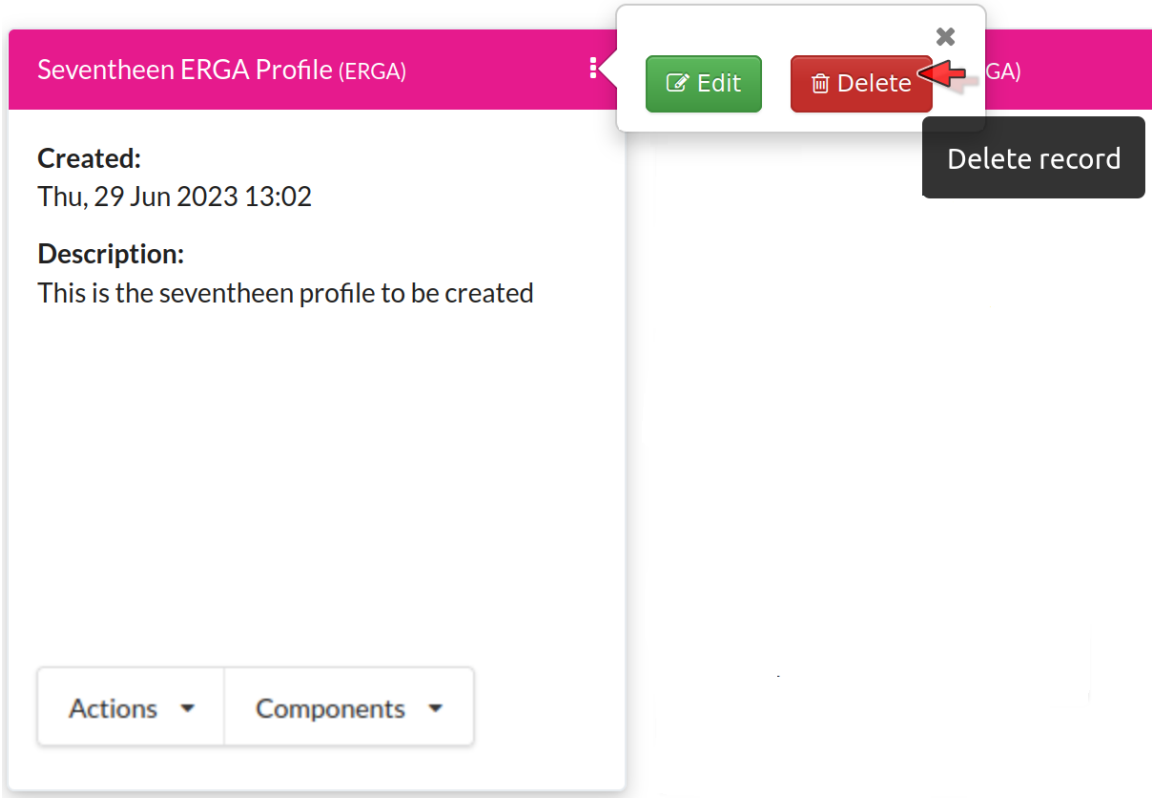


Fig. 4: **COPO Profile: Click the `Delete` button**

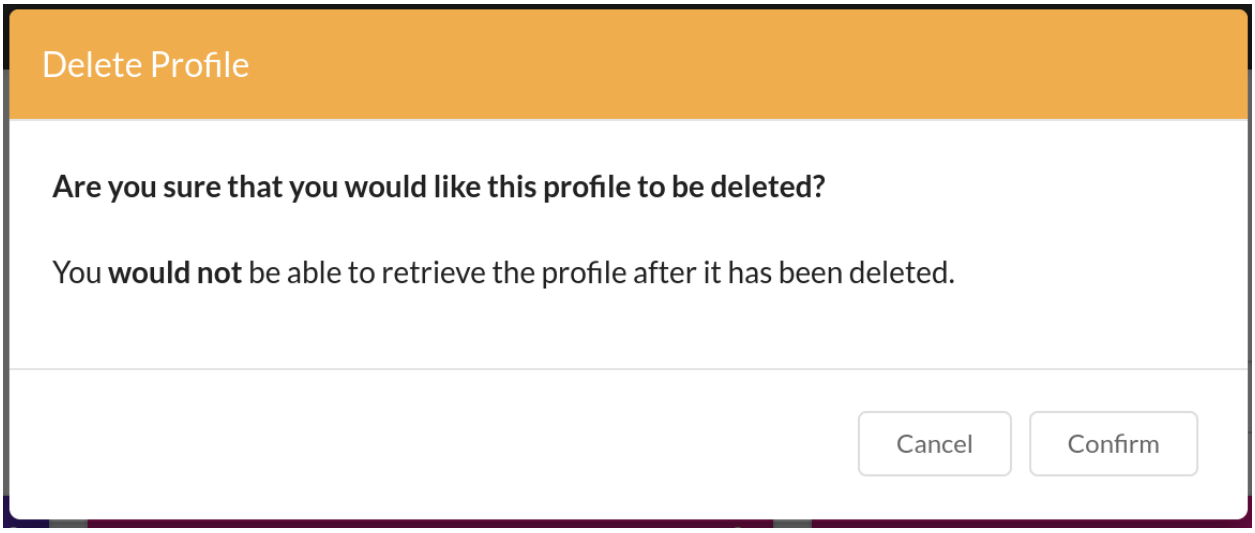


Fig. 5: **COPO Profile: A confirmation dialog will appear. Click the `Confirm` button to confirm the deletion.**

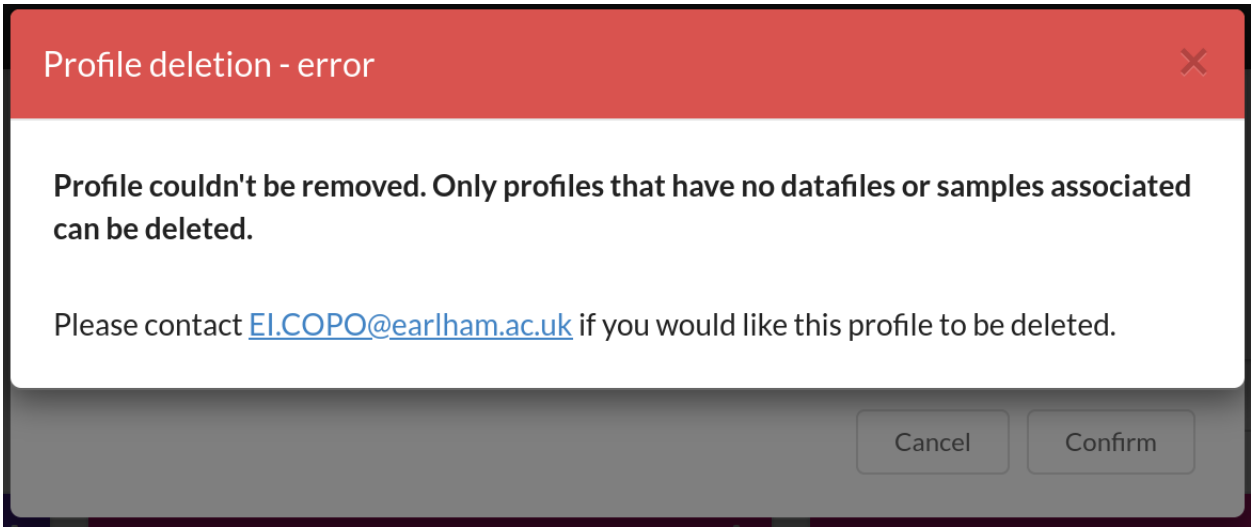


Fig. 6: **COPO Profile: An error will occur if you try to remove a profile that that has associated samples or datafiles.**

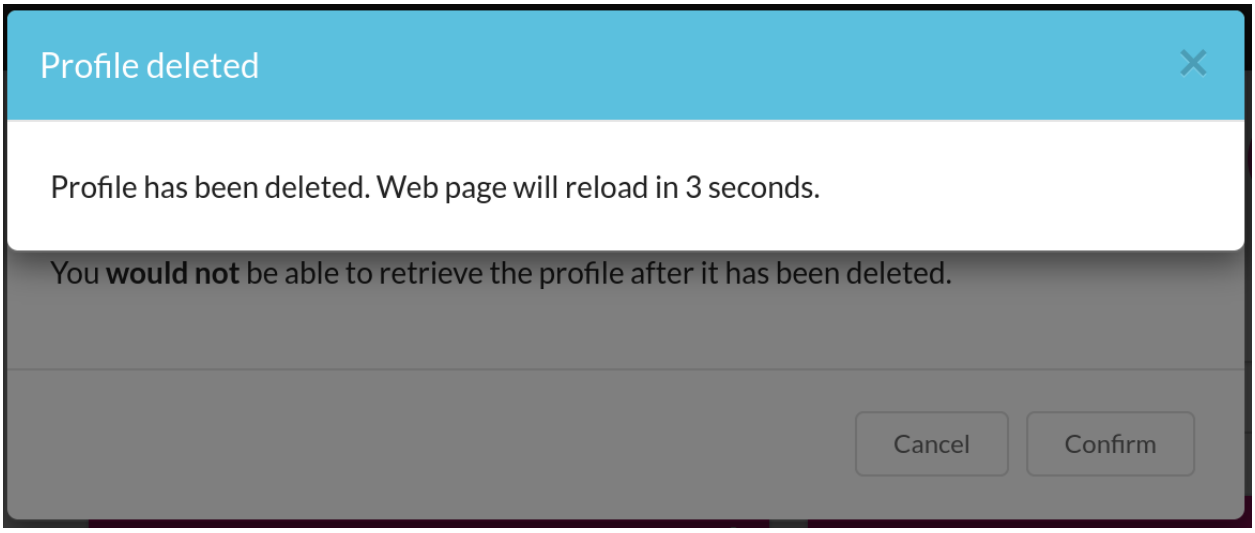


Fig. 7: **COPO Profile: Profile record has been deleted. Wait 3 seconds for the web page to be automatically reloaded.**

CHAPTER

TWENTY

UPDATING SAMPLES

Samples can be updated by reuploading/resubmitting the manifest that was used to create them with the amended changes **before** or **after** the samples have been accepted by a sample manager and submitted to ENA (European Nucleotide Archive)[1].

Please note that not all field values of existing samples can be updated via the manifest resubmission update method. The fields that can be updated with that method are listed below. Those fields are known as *non-compliance fields* while the fields that are not listed or cannot be updated are known as *compliance fields*[2].

See the *How to Download Submitted Sample Manifest* section for guidelines about how to download a manifest that was already submitted.

Hint: To view a list of fields that can be updated for a particular project, collapse the list of fields by clicking the ▶ button below.

Important: The amended manifest or updated spreadsheet file **must** be uploaded to the same profile that was initially used to upload the manifest (before any modifications were done).

An error will occur if you perform one of the following actions:

- Upload the amended manifest to a different profile[3] (other than the one used to upload the manifest to initially)
- Upload the amended manifest to a new profile
- Delete a profile that already has samples associated with it

---

Note:

- The manifest must be uploaded again with the amendments included for the change or update to occur.
- You **cannot** reupload a manifest with some of the existing samples removed. The removed samples (from the reuploaded manifest) will not be automatically removed from the manifest sample record. If you would like existing samples to be removed or deleted from the manifest record, please contact for assistance detailing the profile title and reason for the removal.
- Fields are also referred to as columns in the manifest.

---

[1] See term: *ENA*.
[2] See term: *Compliance field*.
[3] Also known as COPO profile. See term: *COPO profile*.

- BARCODE_HUB
- BARCODE_PLATE_PRESERVATIVE
- COLLECTOR_SAMPLE_ID
- CULTURE_OR_STRAIN_ID
- DATE_OF_PRESERVATION
- DEPTH
- DIFFICULT_OR_HIGH_PRIORITY_SAMPLE
- ELEVATION
- HAZARD_GROUP
- IDENTIFIED_BY
- IDENTIFIED_HOW
- IDENTIFIER_AFFILIATION
- INFRASPECIFIC_EPITHET
- LIFESTAGE
- PARTNER_SAMPLE_ID
- PLATE_ID_FOR_BARCODING
- PRESERVED_BY
- PRESERVER_AFFILIATION
- PURPOSE_OF_SPECIMEN
- RELATIONSHIP
- SEX
- SIZE_OF_TISSUE_IN_TUBE
- SPECIMEN_ID_RISK
- TIME_ELAPSED_FROM_COLLECTION_TO_PRESERVATION
- TIME_OF_COLLECTION
- TISSUE_FOR_BARCODING
- TISSUE_REMOVED_FOR_BARCODING
- TUBE_OR_WELL_ID_FOR_BARCODING
- VOUCHER_ID

- BARCODE_HUB
- BARCODE_PLATE_PRESERVATIVE
- COLLECTOR_SAMPLE_ID
- CULTURE_OR_STRAIN_ID
- DATE_OF_PRESERVATION
- DEPTH
- DIFFICULT_OR_HIGH_PRIORITY_SAMPLE
- ELEVATION
- GAL_SAMPLE_ID
- HAZARD_GROUP
- IDENTIFIED_BY
- IDENTIFIED_HOW
- IDENTIFIER_AFFILIATION
- INFRASPECIFIC_EPITHET
- LIFESTAGE
- PLATE_ID_FOR_BARCODING
- PRESERVED_BY
- PRESERVER_AFFILIATION
- PURPOSE_OF_SPECIMEN
- RELATIONSHIP
- SEX
- SIZE_OF_TISSUE_IN_TUBE
- SPECIMEN_IDENTITY_RISK
- TIME_ELAPSED_FROM_COLLECTION_TO_PRESERVATION
- TIME_OF_COLLECTION
- TISSUE_FOR_BARCODING
- TISSUE_REMOVED_FOR_BARCODING
- TUBE_OR_WELL_ID_FOR_BARCODING
- VOUCHER_ID

- ASSOCIATED_TRADITIONAL_KNOWLEDGE_CONTACT
- ASSOCIATED_TRADITIONAL_KNOWLEDGE_OR_BIOCULTURAL_PROJECT_ID
- ASSOCIATED_TRADITIONAL_KNOWLEDGE_OR_BIOCULTURAL_RIGHTS_HELD_BY
- BARCODE_HUB
- BARCODING_STATUS
- BARCODE_PLATE_PRESERVATIVE
- BIOBANKED_TISSUE_PRESERVATIVE
- COLLECTED_BY
- COLLECTION_LOCATION
- COLLECTOR_AFFILIATION
- COLLECTOR_SAMPLE_ID
- COMMON_NAME
- CULTURE_OR_STRAIN_ID
- DATE_OF_COLLECTION
- DATE_OF_PRESERVATION
- DECIMAL_LATITUDE
- DECIMAL_LONGITUDE
- DEPTH
- DESCRIPTION_OF_COLLECTION_METHOD
- DIFFICULT_OR_HIGH_PRIORITY_SAMPLE
- DNA_REMOVED_FOR_BIOBANKING
- DNA_VOUCHER_ID_FOR_BIOBANKING
- ETHICS_PERMITS_DEF
- ETHICS_PERMITS_HOLDER
- ETHICS_PERMITS_REQUIRED
- FAMILY
- GAL
- GAL_SAMPLE_ID
- GENUS
- GRID_REFERENCE
- HABITAT
- HAZARD_GROUP
- IDENTIFIED_BY
- IDENTIFIED_HOW
- IDENTIFIER_AFFILIATION
- IDENTIFIER_AFFILIATION
- INDIGENOUS_RIGHTS_APPLICABLE
- INDIGENOUS_RIGHTS_DEF
- INDIGENOUS_RIGHTS_DEF
- INFRASPECIFIC_EPITHET

- LIFESTAGE
- NAGOYA_PERMITS_DEF
- NAGOYA_PERMITS_FILENAME
- NAGOYA_PERMITS_REQUIRED
- ORDER_OR_GROUP
- ORGANISM_PART
- ORIGINAL_COLLECTION_DATE
- ORIGINAL_GEOGRAPHIC_LOCATION
- OTHER_INFORMATION
- PRESERVATION_APPROACH
- PRESERVATIVE_SOLUTION
- PRESERVED_BY
- PRESERVER_AFFILIATION
- PROXY_TISSUE_VOUCHER_ID_FOR_BIOBANKING
- PROXY_VOUCHER_ID
- PROXY_VOUCHER_LINK
- PURPOSE_OF_SPECIMEN
- REGULATORY_COMPLIANCE
- RELATIONSHIP
- SAMPLE_COORDINATOR
- SAMPLE_COORDINATOR_AFFILIATION
- SAMPLE_COORDINATOR_ORCID_ID
- SAMPLING_PERMITS_FILENAME
- SAMPLING_PERMITS_REQUIRED
- SCIENTIFIC_NAME
- SEX
- SIZE_OF_TISSUE_IN_TUBE
- SPECIMEN_IDENTITY_RISK
- TAXON_ID
- TAXON_REMARKS
- TIME_ELAPSED_FROM_COLLECTION_TO_PRESERVATION
- TIME_OF_COLLECTION
- TISSUE_FOR_BARCODING
- TISSUE_FOR_BIOBANKING
- TISSUE_REMOVED_FOR_BARCODING
- TISSUE_REMOVED_FOR_BIOBANKING
- TISSUE_REMOVED_FROM_BARCODING
- TISSUE_VOUCHER_ID_FOR_BIOBANKING
- TUBE_OR_WELL_ID_FOR_BARCODING
- VOUCHER_ID
- VOUCHER_INSTITUTION
- VOUCHER_LINK

CHAPTER

TWENTYONE

UPDATING SEQUENCE ANNOTATIONS

This section describes how to update **Sequence Annotations**[1]. The **Sequence Annotations** update process is similar to the **Sequence Annotations** submission process. Therefore, please refer to the *Sequence Annotations Submission* section for more information.

Note:

- Sequence Annotations must be uploaded before they can be updated.

See also:

- *How to Submit Sequence Annotations*
- *How to Delete Sequence Annotations*

1. Click the desired sequence annotation from the table of uploaded sequence annotations to highlight it.

   Then, click the **Edit** button to update or edit the annotation as shown below:

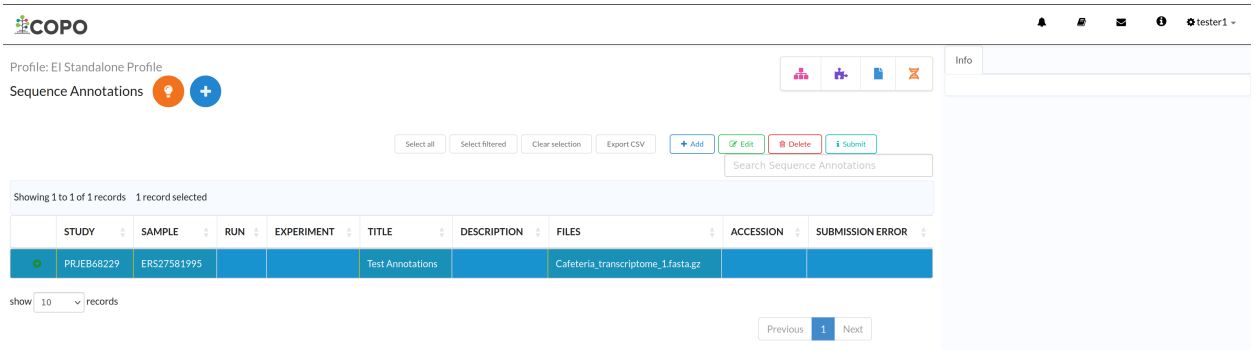


Fig. 1: **Sequence Annotations' update: Click 'Edit' button after having clicked the desired sequence annotation**

2. An **Add Sequence Annotation** dialogue is displayed. Provide the updated details then, click the **Submit Annotation** button.

3. The updated sequence annotation(s) will be displayed on the **Sequence Annotations** web page after a successful update.

[1] See: *Sequence Annotation*.

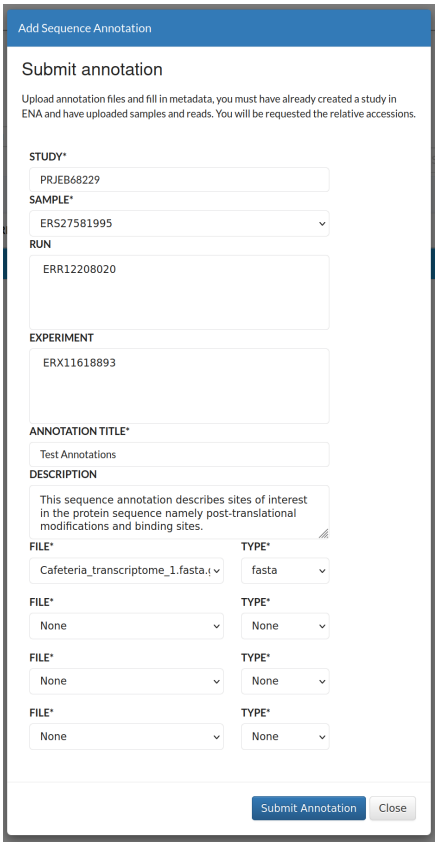


Fig. 2: **Sequence Annotations' update: 'Submit Annotation' dialogue with updated information**

Hint: Sequence annotation records that are highlighted **yellow** indicate that the records are pending submission. The records will be highlighted **green** after a successful automatic submission.

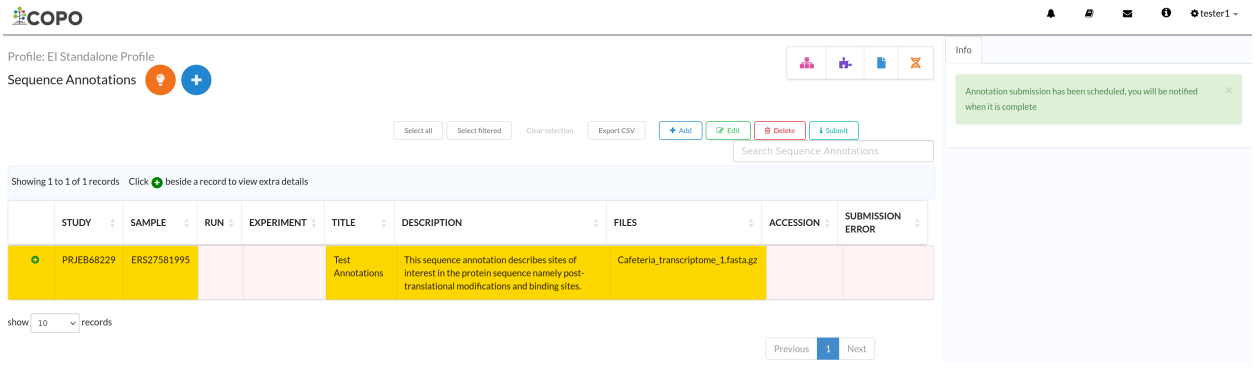
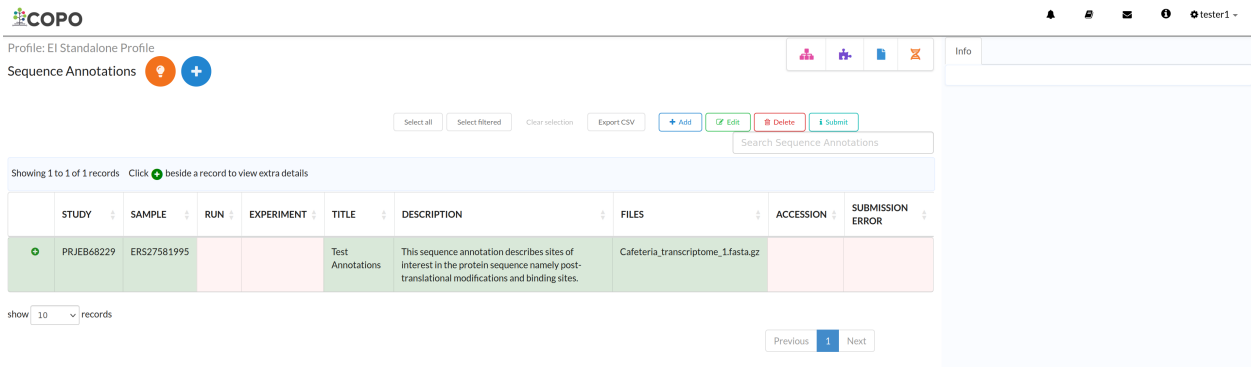


Fig. 3: **Sequence Annotations' update: Sequence Annotations' web page displaying the (pending) updated sequence annotation(s)**



Sequence Annotations' update: Sequence Annotations' web page displaying the updated sequence annotation(s) after it has been submitted

CHAPTER

TWENTYTWO

UPDATING READS

This section describes how to update **Reads** for a given checklist. The **Reads** update process is similar to the **Reads** submission process. Therefore, please refer to the *Reads Submission* section for more information.

Note:

- Reads must be uploaded before they can be updated.

See also:

- *How to Submit Reads*
- *Explore Various Types of Reads Checklist*
- *How to Submit Files*

1. On the **Reads'** web page, click the checklist dropdown then, choose a desired checklist from the list of available checklists that you would like to update a **Reads** for.

   Then, click the button as shown below to update a Reads for the chosen checklist as shown below:

   Note: The colour of the button is based on the type of profile that you are updating a Reads for.

   See the *Profile Types Legend* section regarding the colour code for the various types of project profiles on COPO.

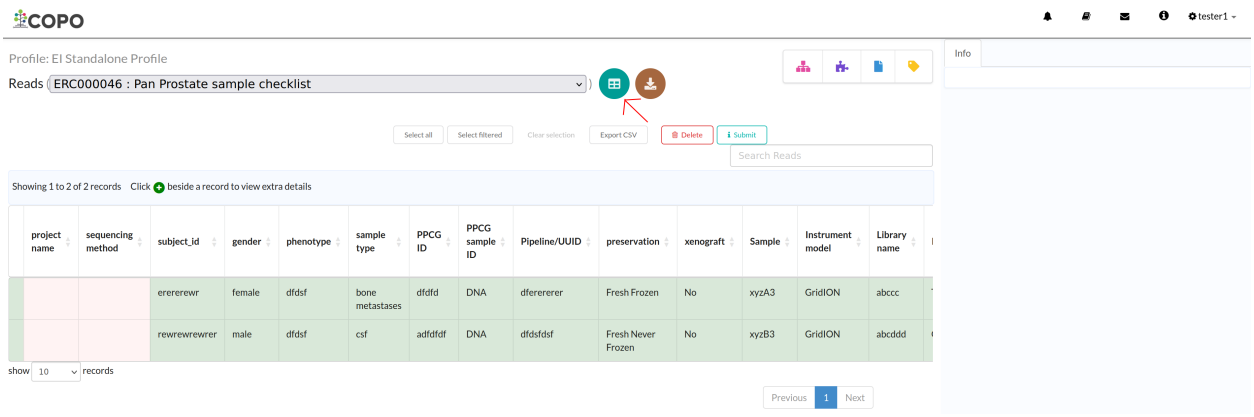


Fig. 1: **Reads' update: Click 'Add Reads' from Spreadsheet' button to upload an amended reads manifest**

2. An **Upload Read Manifest** dialogue is displayed. Click the Upload Read Manifest button to choose an updated file from your local system.

   The uploaded manifest is shown in a table in the **Upload Read Manifest** dialogue as shown below. Click the Finish button to complete the update process of the reads manifest.

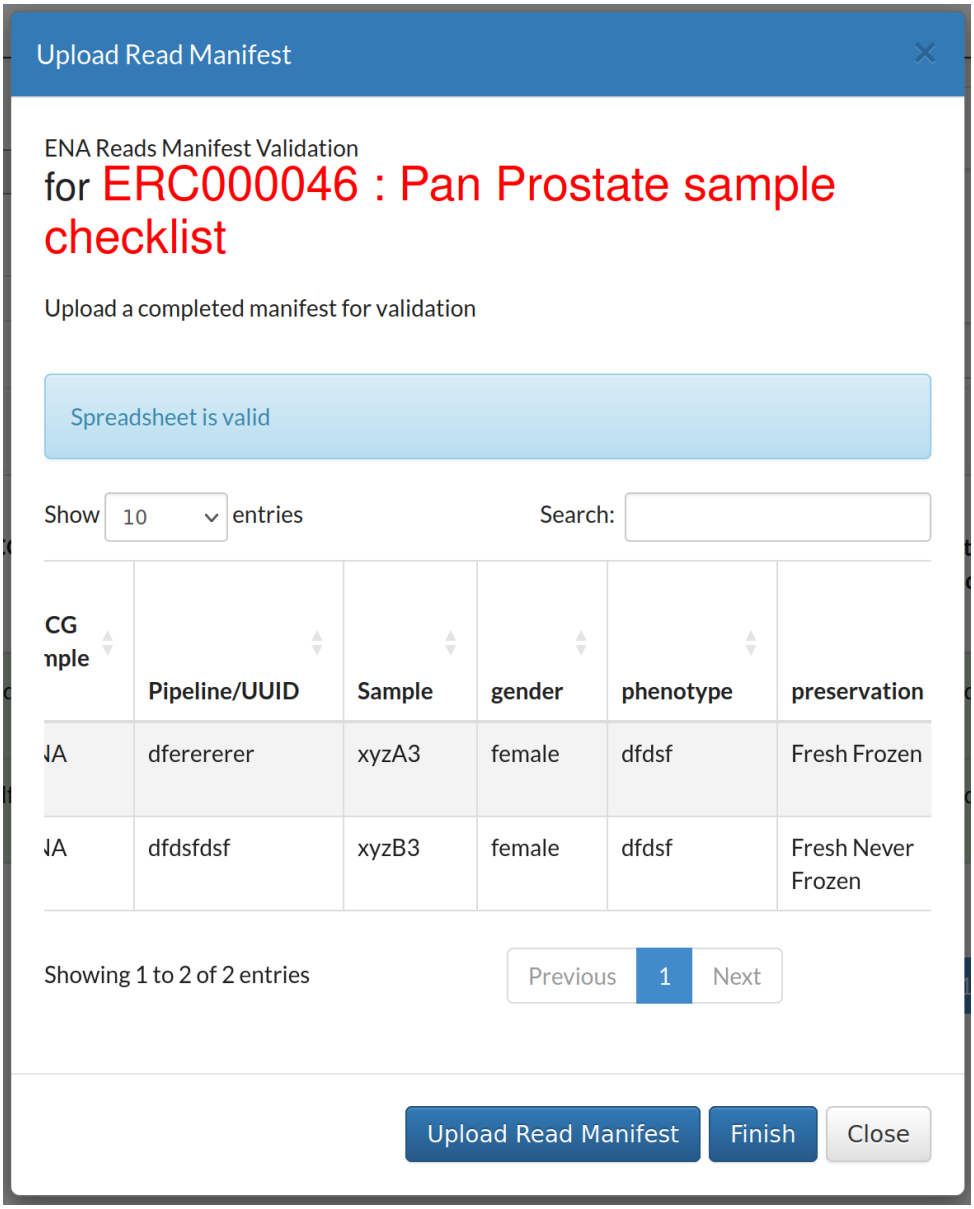


Fig. 2: **Reads' update: 'Upload Read Manifest' dialogue with updated manifest**

3. The update read(s) will be displayed on the **Reads** web page after a successful update.

Hint: Reads records that are highlighted **yellow** indicate that the records are pending submission. The records will be highlighted **green** after a successful submission.

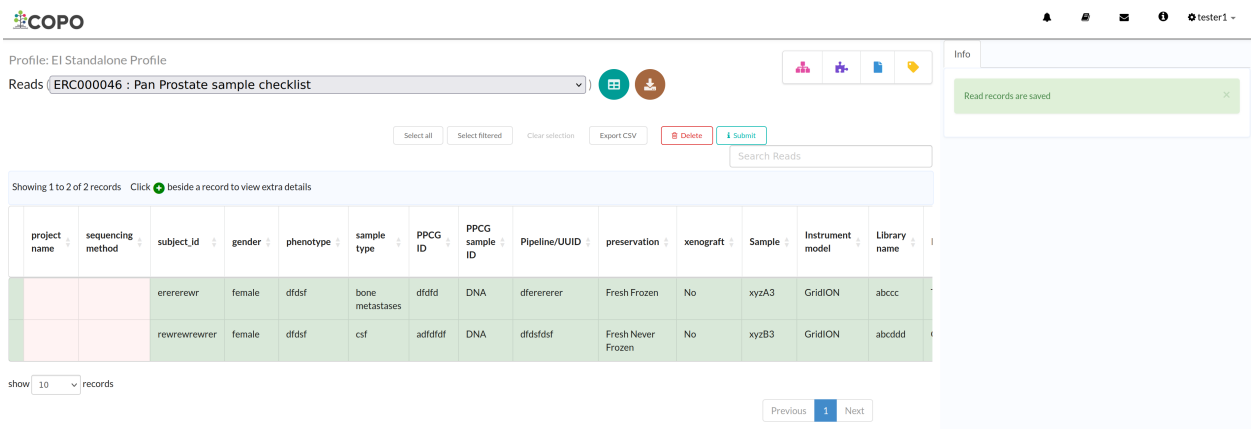


Fig. 3: **Reads' update: Reads' web page displaying the updated read(s)**

CHAPTER

TWENTYTHREE

ADDING CHILDREN PROJECTS TO PARENT PROJECTS

23.1 In COPO

See *Choose associated profile type(s)* section when creating a profile for more details.

23.2 In Public Repositories

Please contact the if you would like child projects (subprojects) to be added to parent (main) projects in public repositories like ENA[1] or NCBI[2].

Please provide the parent project accession and the child project accession(s) when making the email request.

[1] See term: *ENA*.
[2] See term: *NCBI*.

CHAPTER

TWENTYFOUR

DOWNLOADING MANIFEST TEMPLATES & STANDARD OPERATING PROCEDURES (SOPS)

In COPO, *manifests* refer to Microsoft Excel spreadsheets that are used to record metadata about a biological research.

24.1 Downloading Manifest Templates

Blank manifest templates are available for download below or on the Manifests web page.

In addition, click the button located on the COPO **Samples** web page to download a manifest template. See samples-component component for more details about accessing the **Samples** web page.

`ASG manifest template`[1]

`DToL manifest template`[2]

`ERGA manifest template`[3]

`Barcoding manifest template (within Tree of Life profiles)`

`Reads manifest template (within Tree of Life profiles)`

Note:

- The manifest templates for reads within Stand-alone profiles are available for download on the **Reads web page** for those profiles. See *Reads Submission* section for more details.
- Barcoding manifest templates are available for download on the Barcoding manifest web page. See *Barcoding Manifest Submission* section for more details.

---

[1] See term: *ASG (Aquatic Symbiosis Genomics)*.
[2] See term: *DToL (Darwin Tree of Life)*.
[3] See term: *ERGA (European Reference Genome Atlas)*.

24.2 Recording Data in Manifests

Each manifest follow a manual known as *Standard Operating Procedure (SOP)*[4] which provides a detailed guidance on how to record metadata in the manifest.

The SOP for the various manifests are available below.

In addition, click the button located on the COPO **Samples** web page to download the SOP. See samples-component component for more details about accessing the **Samples** web page.

24.2.1 ASG SOP

This follows most of the instructions outlined in the DToL SOP, with some minor changes and omissions.

`ASG SOP`

ASG Project

24.2.2 DToL (Darwin Tree of Life) SOP

`DToL SOP`

DToL SOP GitHub repository

DToL Project

24.2.3 DToL_ENV (Darwin Tree of Life Environmental Samples) SOP

This will be made available soon.

24.2.4 ERGA SOP

`ERGA SOP`

ERGA SOP GitHub repository

24.2.5 POP_GENOMICS (Population Genomics) SOP

`POP_GENOMICS SOP`

ERGA SOP GitHub repository

See also:

- *Using manifest wizard to prefill manifests*

[4] See term: *SOP*.

CHAPTER

TWENTYFIVE

USING COPO API

The COPO API is a RESTful website service that allows users to interact with the COPO system. The API is built using the Django REST (REpresentational State Transfer)[1] Framework.

Hint: The button,

, indicates that the API method endpoints will produce results from the website host server i.e. if the live COPO website is used to query the endpoint, then, the live results will be retrieved while the opposite occurs if one uses the demo website to query the API method endpoints.

25.1 API End-points

The COPO API (Application Programming Interface)[2] comprises manifest, sample, statistics and profile endpoints. Their results are available for download in csv, json or ro-crate[3] file formats depending on the record type as shown in the table below.

Each endpoint results contain metadata provided by the submitter.

Table 1: Available Result Formats for Records

File Format	Available Record Types
csv	Study, Sample, Manifest
json	Study, Sample, Manifest
ro-crate	Sample

Note: The live COPO website API endpoints are used in the examples below.

To use the demo website API endpoints, replace `https://copo-project.org/api/` with `https://demo.copo-project.org/api/` in the examples below.

[1] See term: *REST*
[2] See term: *API*
[3] See term: *RO-Crate*

25.1.1 Audit Endpoints

Fetch Sample Updates Between Dates

```
https://copo-project.org/api/audit/sample/{from}/{to}
```

OR

```
$ curl -X GET "https://copo-project.org/api/audit/sample/{from}/{to}" -H  "accept:␣
↪application/json"
```

This results in a list of sample updates that occurred between a given date period.

Fetch Sample Updates by Manifest ID

```
https://copo-project.org/api/audit/sample/manifest_id/{manifest_id}
```

OR

```
$ curl -X GET "https://copo-project.org/api/audit/sample/manifest_id/{manifest_id}" -H
↪"accept: application/json"
```

This results in a list of sample updates by manifest IDs (Identifications)[4].

Fetch Sample Updates by COPO ID

```
https://copo-project.org/api/audit/sample/{copo_id}
```

OR

```
$ curl -X GET "https://copo-project.org/api/audit/sample/copo_id/{copo_id}" -H  "accept:␣
↪application/json"
```

This results in a list of sample updates based on `{copo_id}`.

Fetch Sample Updates by Update Type

```
https://copo-project.org/api/audit/sample/update_type/{update_type}
```

OR

```
$ curl -X GET "https://copo-project.org/api/audit/sample/update_type/{update_type}" -H
↪"accept: application/json"
```

This results in a list of sample updates based who performed the update. The `update_type` can be **system** or **user**.

A **system** update occurs when the update was performed by COPO while a **user** update occurs when a user reuploads a manifest with amended sample metadata.

Please note that not all sample information that has been uploaded already can be updated when the manifest is reuploaded. Only fields that are updatable are updated when a manifest is reuploaded.

[4] See term: *Manifest ID*

See the *Updating Samples* section for more information as well as the *Fetch updatable fields by project* API method.

Fetch Sample Updates by Field and Field Value

```
https://copo-project.org/api/audit/sample/{field}/{field_value}
```

OR

```
$ curl -X GET "https://copo-project.org/api/audit/sample/{field}/{field_value}" -H
↪"accept: application/json"
```

This results in a list of sample updates based on a sample field value and one of the following sample fields:

- RACK_OR_PLATE_ID
- SPECIMEN_ID
- TUBE_OR_WELL_ID
- biosampleAccession
- public_name
- sraAccession

Fetch Sample Updates by ASG Sample Type

```
https://copo-project.org/api/audit/sample/asg
```

OR

```
$ curl -X GET "https://copo-project.org/api/audit/sample/asg" -H  "accept: application/
↪json"
```

This results in a list of updates for ASG[8]. sample types.

Fetch Sample Updates by DTOL Sample Type

```
https://copo-project.org/api/audit/sample/dtol
```

OR

```
$ curl -X GET "https://copo-project.org/api/audit/sample/dtol" -H  "accept: application/
↪json"
```

This results in a list of updates for DToL[9]. sample types.

[8] See term: *DToL*
[9] See term: *ENA*

Fetch Sample Updates by ERGA Sample Type

```
https://copo-project.org/api/audit/sample/erga
```

OR

```
$ curl -X GET "https://copo-project.org/api/audit/sample/erga" -H  "accept: application/
↪json"
```

This results in a list of updates for ERGA[10]. sample types.

25.1.2 Manifest Endpoints

Fetch Manifests

```
https://copo-project.org/api/manifest
```

OR

```
$ curl -X GET "https://copo-project.org/api/manifest" -H  "accept: application/json"
```

This results in a list of all manifest IDs[Page 128, 4]. The manifest identification can be used to retrieve records in the other endpoints.

Fetch Current Manifest Versions

```
https://copo-project.org/api/manifest/current_version
```

OR

```
$ curl -X GET "https://copo-project.org/api/manifest/current_version" -H  "accept:␣
↪application/json"
```

This displays the current or latest manifest version of each manifest project brokered through COPO.

Fetch Sample Records by Manifest ID

```
https://copo-project.org/api/manifest/{manifest_id}
```

OR

```
$ curl -X GET "https://copo-project.org/api/manifest/{manifest_id}" -H  "accept:␣
↪application/json"
```

This results in a list of sample records for the given manifest IDs[Page 128, 4].

[10] See term: *Sample manager*

Fetch Manifests between Dates

```
https://copo-project.org/api/manifest/{from}/{to}
```

OR

```
$ curl -X GET "https://copo-project.org/api/manifest/{from}/{to}" -H  "accept:␣
↪application/json"
```

This results in a list of all manifest IDs[Page 128, 4] recorded in the given date period.

Fetch Manifests between Dates for a Project

```
https://copo-project.org/api/manifest/{project}/{from}/{to}
```

OR

```
$ curl -X GET "https://copo-project.org/api/manifest/{project}/{from}/{to}" -H  "accept:␣
↪application/json"
```

This results in a list of all manifest IDs[Page 128, 4] recorded in the given date period for a given project.

Fetch Sample Record Status in a Manifest

```
https://copo-project.org/api/manifest/{manifest_id}/sample_statuses
```

OR

```
$ curl -X GET "https://copo-project.org/api/manifest/{manifest_id}/sample_statuses" -H
↪"accept: application/json"
```

This results in minimal sample status information for each sample contained in the given `{manifest_id}`.

Validate Manifest by Profile ID

Note: Authentication is required in order to use this API method. Obtain an API token from the following endpoint before using this method:

```
https://copo-project.org/api/manifest/validate
```

This results in the ID of the validation report to be queried.

Validate Manifest by Report ID

Note: Authentication is required in order to use this API method. Obtain an API token from the following endpoint before using this method:

```
https://copo-project.org/api/manifest/validate/report/
```

OR

```
$ curl -X POST "https://copo-project.org/api/manifest/validate/report/" -H  "accept: */*
↪" -H  "Content-Type: application/x-www-form-urlencoded" -d "validation_report_id=
↪{report-id}
```

This gives the status and/or validation errors for a manifest based on the manifest report ID.

Validate Manifest

Note: Authentication is required in order to use this API method. Obtain an API token from the following endpoint before using this method:

```
https://copo-project.org/api/apiKey
```

```
https://copo-project.org/api/manifest/validations
```

OR

```
$ curl -X POST "https://copo-project.org/api/manifest/validations/" -H  "accept: */*" -d
↪""
```

This checks whether a given manifest passes or fails validation for the authorised user.

25.1.3 Sample Endpoints

Fetch Sample Records by Project

```
https://copo-project.org/api/sample/{project}
```

OR

```
$ curl -X GET "https://copo-project.org/api/sample/{project}" -H  "accept: application/
↪json"
```

This results in a list of all the samples of a project in COPO.

Fetch Sample Records by Associated Project Type

Hint: The associated project type is the project type that the sample is subproject of. For example, a sample may be associated with a project type of "BGE" but the sample itself may be an "ERGA" sample.

In sample records, the associated project type is referred to as **associated_tol_project** whereas in profile records, it is referred to as **associated_type**.

```
https://copo-project.org/api/sample/associated_tol_project/{values}
```

OR

```
$ curl -X GET "https://copo-project.org/api/sample/sample/associated_tol_project/{values}
↪" -H  "accept: application/json"
```

This results in a list of all sample records of a given associated project type(s) in COPO.

Fetch Sample Records between Dates

```
https://copo-project.org/api/sample/project/manifest_version/fields
```

OR

```
$ curl -X GET "https://copo-project.org/api/sample/project/manifest_version/fields}" -H
↪"accept: application/json"
```

This results in a list of fields of a project for a given manifest version.

Fetch Sample Records between Dates

```
https://copo-project.org/api/sample/{from}/{to}
```

OR

```
$ curl -X GET "https://copo-project.org/api/sample/{from}/{to}" -H  "accept: application/
↪json"
```

This results in a list of all samples recorded in the given date period.

Fetch Updatable Fields by Project

```
https://copo-project.org/api/sample/updatable_fields/{project}
```

OR

```
$ curl -X GET "https://copo-project.org/api/sample/updatable_fields/{project}" -H
↪"accept: */*"
```

This results in list of fields that can be updated when a manifest is reuploaded/resubmitted in COPO based on the given `{project}`.

Fetch Sample Records by COPO ID

Hint:

- Sample records IDs are referred to as `copo_id` in COPO and `alias` in ENA.
- Multiple `copo_id` can be provided as a comma separated list in this endpoint.

```
https://copo-project.org/api/sample/copo_id/{copo_ids}
```

OR

```
$ curl -X GET "https://copo-project.org/api/sample/copo_id/{copo_ids}" -H  "accept:␣
↪application/json"
```

This results in full sample information for the sample records returned from the given `{copo_ids}`.

Fetch Sample Records by Biosample ID

Note:

- Biosample accession IDs are assigned to sample records by ENA[Page 129, 9] after the samples have been approved by a sample manager[Page 130, 10].
- The `biosample_id` is referred to as `biosampleAccession` in COPO and `biosample_id` in ENA.

```
https://copo-project.org/api/sample/biosample_id/{biosample_ids}
```

OR

```
$ curl -X GET "https://copo-project.org/api/sample/biosample_id/{biosample_ids}" -H
↪"accept: application/json"
```

This results in full sample information for the sample records returned from the given `{biosample_ids}`.

Fetch Sample Records by Field and Values

Hint:

- Multiple values can be provided as a comma separated list in this endpoint.

```
https://copo-project.org/api/sample/sample_field/{field}/{values}
```

OR

```
$ curl -X GET "https://copo-project.org/api/sample/sample_field/{field}/{values}" -H
↪"accept: application/json"
```

This results in full sample information for the sample records returned from the given `{field}/{values}`.

Fetch Study Records by Sample ID

Hint:

- Multiple `sample_ids` can be provided as a comma separated list in this endpoint.

```
https://copo-project.org/api/sample/StudyFromSampleAccession/{sample_ids}
```

OR

```
$ curl -X GET "https://copo-project.org/api/sample/StudyFromSampleAccession/{sample_ids}
↪" -H  "accept: application/json"
```

This results in full sample information for the sample records returned from the given `{sample_ids}`.

Fetch Sample Records by Study ID

Hint:

- Multiple `biostudyAccessions` can be provided as a comma separated list in this endpoint.

```
https://copo-project.org/api/sample/SampleFromStudyAccession/{biostudyAccessions}
```

OR

```
$ curl -X GET "https://copo-project.org/api/sample/SampleFromStudyAccession/
↪{biostudyAccessions}" -H  "accept: application/json"
```

This results in full sample information for the sample records returned from the given `{biostudyAccessions}`.

25.1.4 Profile Endpoints

Create Profile Record

Note: Authentication is required in order to use this API method. Obtain an API token from the following endpoint before using this method:

```
https://copo-project.org/api/apiKey
```

```
https://copo-project.org/api/profile//make_profile
```

OR

```
$ curl -X POST "https://copo-project.org/api/profile//make_profile" -H  "accept: */*" -d
↪""
```

This creates a profile record for the authenticated user.

Fetch Profile Records

Note: Authentication is required in order to use this API method. Obtain an API token from the following endpoint before using this method:

```
https://copo-project.org/api/apiKey
```

```
https://copo-project.org/api/profile/get_for_user
```

OR

```
$ curl -X POST "https://copo-project.org/api/profile/get_for_user" -H  "accept: */*" -d "
↪"
```

This results in a list of all profiles for the authenticated user.

25.1.5 Statistics' Endpoints

Fetch Number of COPO Users

```
https://copo-project.org/api/stats/number_of_users
```

OR

```
$ curl -X GET "https://copo-project.org/api/stats/number_of_users" -H  "accept:␣
↪application/json"
```

This results in the total number of registered users in COPO.

Fetch Number of Sample Records by Sample Type

```
https://copo-project.org/api/stats/number_of_samples/{sample_type}
```

OR

```
$ curl -X GET "https://copo-project.org/api/stats/number_of_dtol_samples/{sample_type}" -
↪H  "accept: application/json"
```

This results in the total number of registered sample records in COPO by a given sample type.

Fetch Tree of Life (ToL) Projects Brokered by COPO

```
https://copo-project.org/api/stats/tol_projects
```

OR

```
$ curl -X GET "https://copo-project.org/api/stats/tol_projects" -H  "accept: application/
↪json"
```

This results in a list of all main/primary projects brokered by COPO.

Fetch Associated Tree of Life (ToL) Projects Brokered by COPO

```
https://copo-project.org/api/stats/associated_tol_projects
```

OR

```
$ curl -X GET "https://copo-project.org/api/stats/associated_tol_projects" -H  "accept:␣
↪application/json"
```

This results in a list of all child projects/ subprojects brokered by COPO.

CHAPTER

TWENTYSIX

EXPLORING MANIFEST WIZARD

The manifest wizard is a tool to help you populate a manifest with common data values. It is a simple wizard that will ask a few questions then, generate the manifest file for you.

The manifest wizard can be accessed via the **Prefilled template** download dropdown menu option on the Manifests web page.

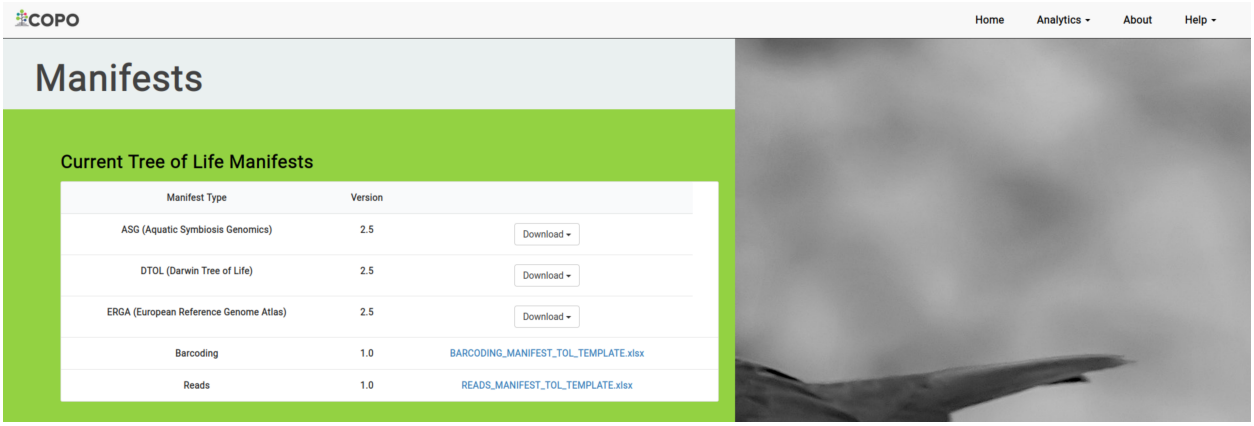


Fig. 1: **Manifests web page: Tabular information for various manifest types and their current versions**

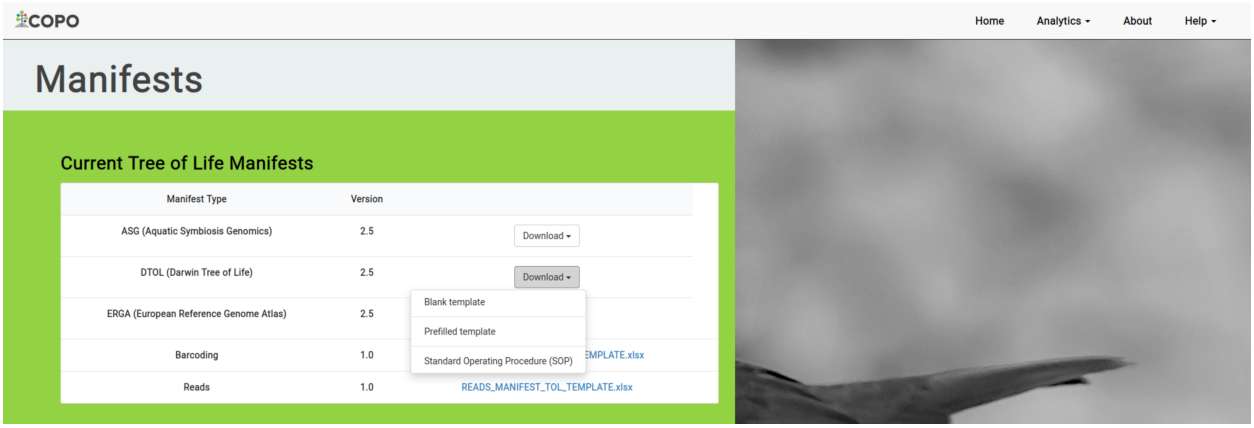


Fig. 2: **Manifests web page: Manifest download template options -** `Blank template` **and** `Prefilled template` with option to download, `Standard Operating Procedure (SOP)`

Blank manifest templates can be downloaded via the **Blank template** dropdown menu option or from the *Download Manifest Templates* section.

26.1 Prefill Manifests

1. Navigate to the Manifests web page. Then, click the **Prefilled template** option from the dropdown menu.

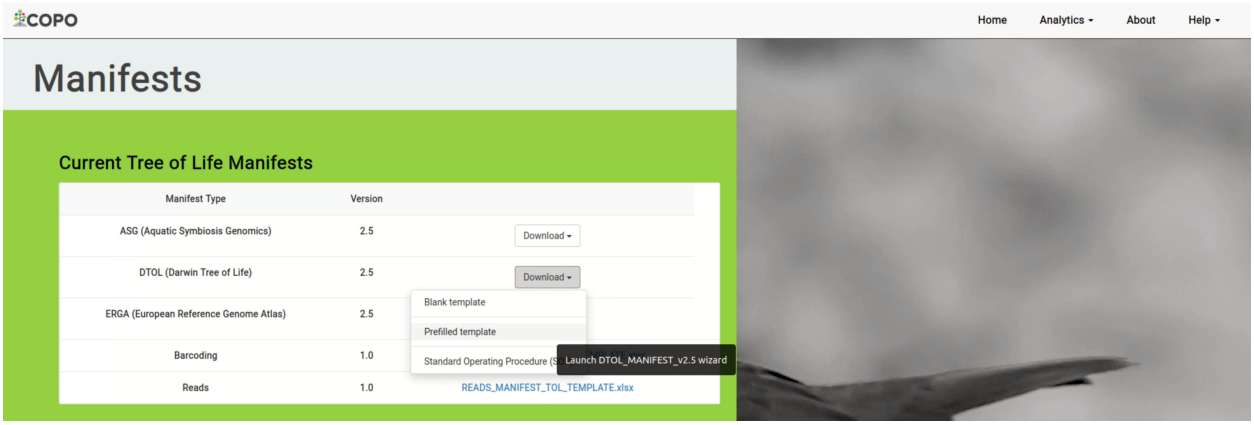


Fig. 3: **Manifests web page: Click 'Prefilled template' dropdown menu option**

2. In the **Tree of Life Manifest Wizard** displayed, enter the number samples that you would like to describe then, input a common value for a field by selecting the field name from the dropdown list.

Note: The manifest wizard will only display fields that are associated with the type of manifest that you would like to prefill.

Once you have inputted a value for a common field, that value will be prefilled according to the number of samples that you have inputted.

---

3. Click the Next → to proceed to the next step of the manifest wizard.
4. Click the Finish button to download the prefilled manifest file.
5. Preview of downloaded manifest with prefilled metadata

See also:

- *Standard Operating Procedure (SOP) guidelines* for manifests
- *Download Manifest Templates*

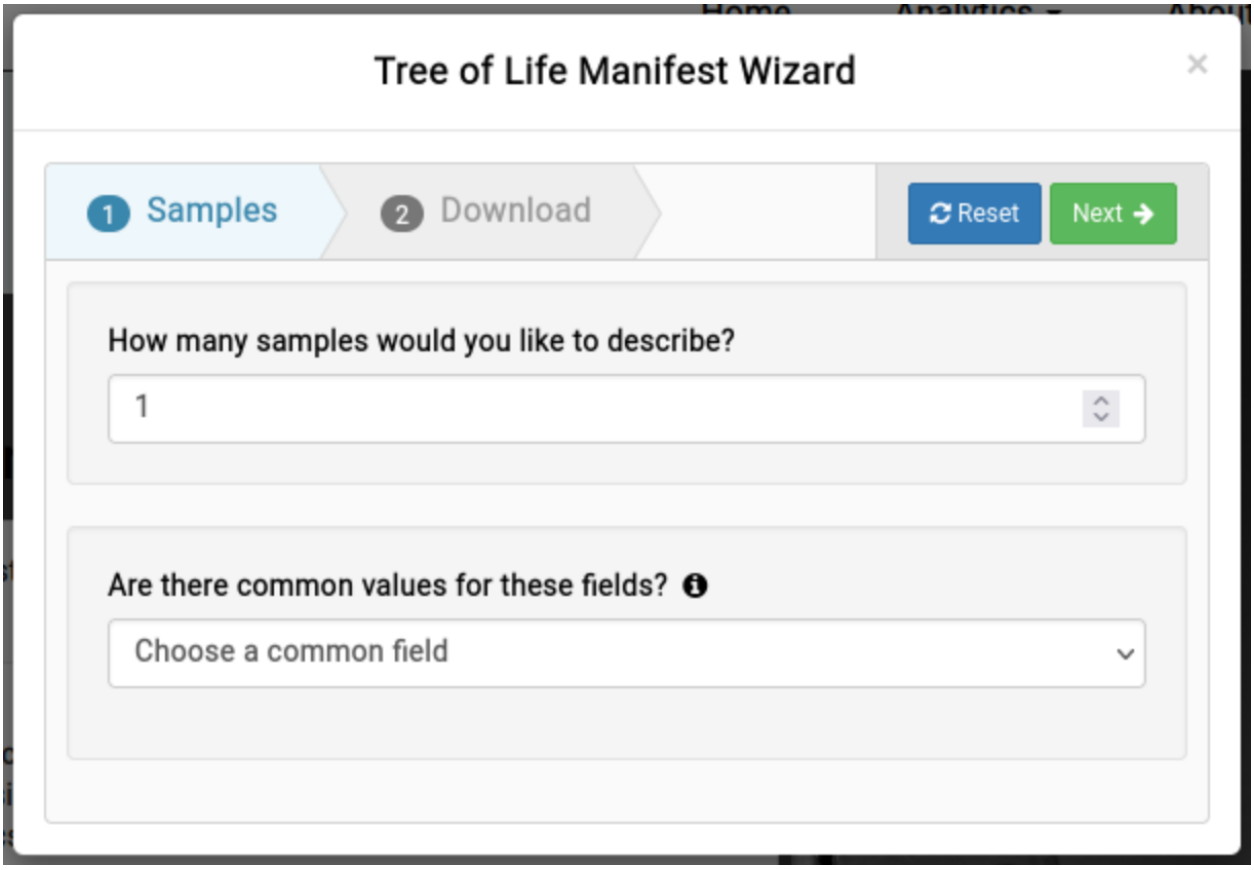


Fig. 4: **Manifests web page: Tree of Life Manifest wizard**

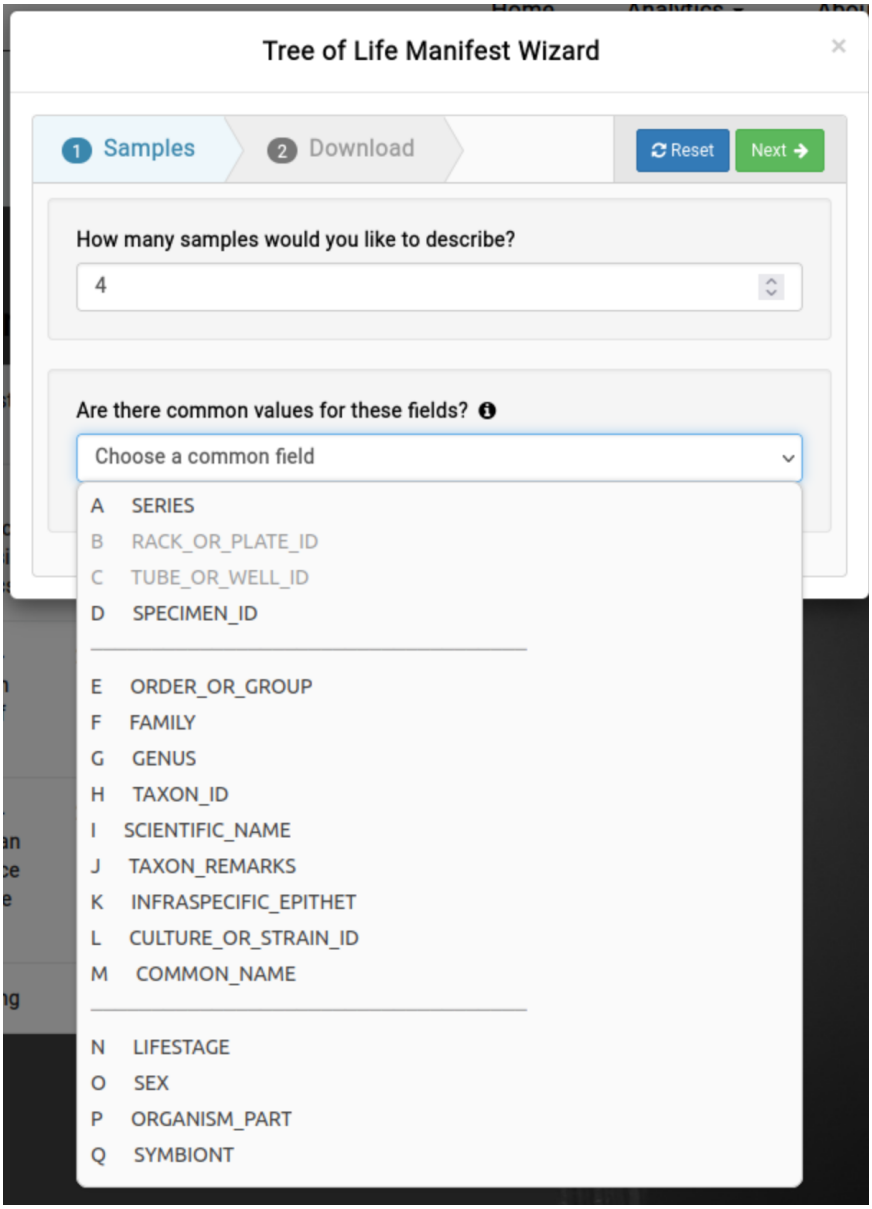


Fig. 5: **Tree of Life Manifest wizard: Common field dropdown menu**

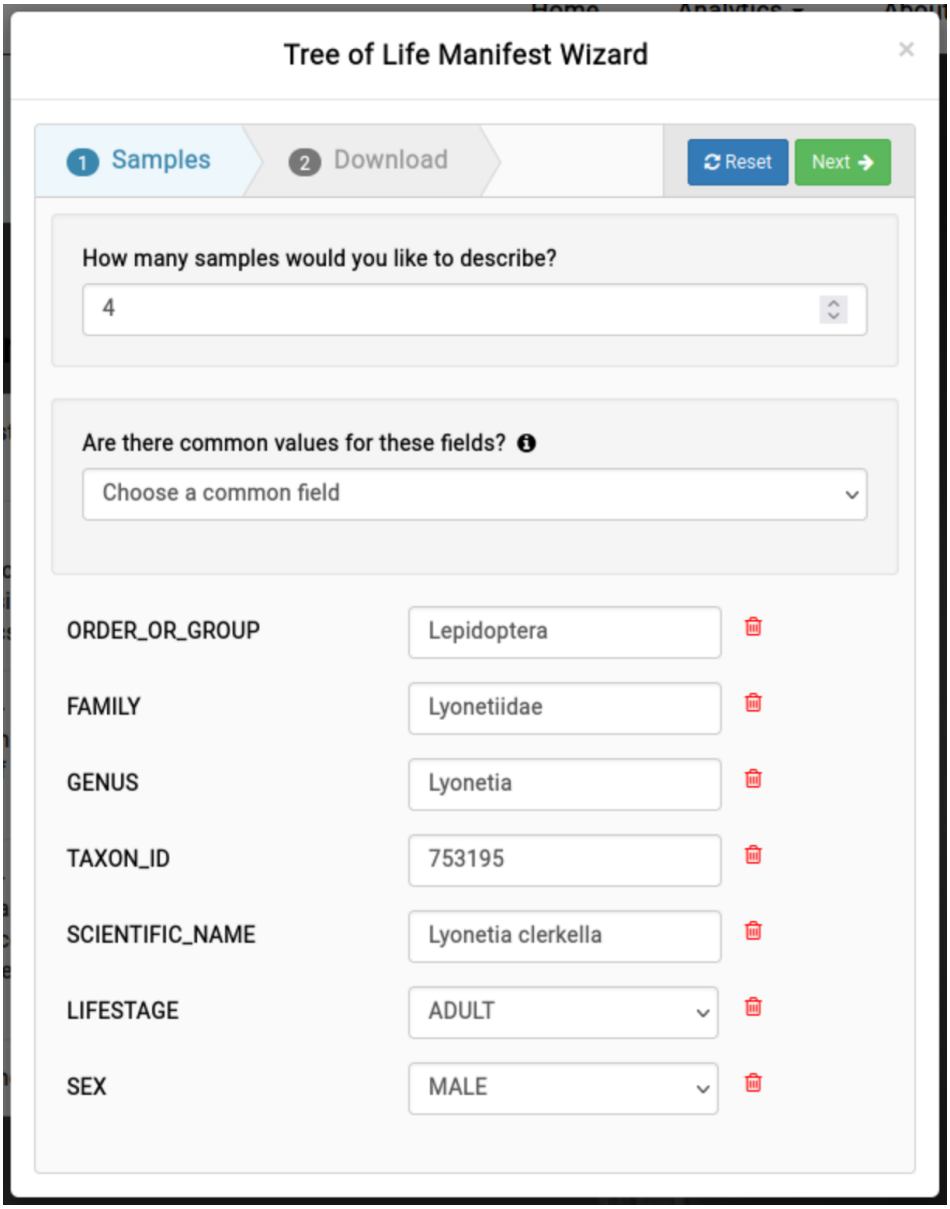


Fig. 6: **Tree of Life Manifest wizard: Inputted common field values**

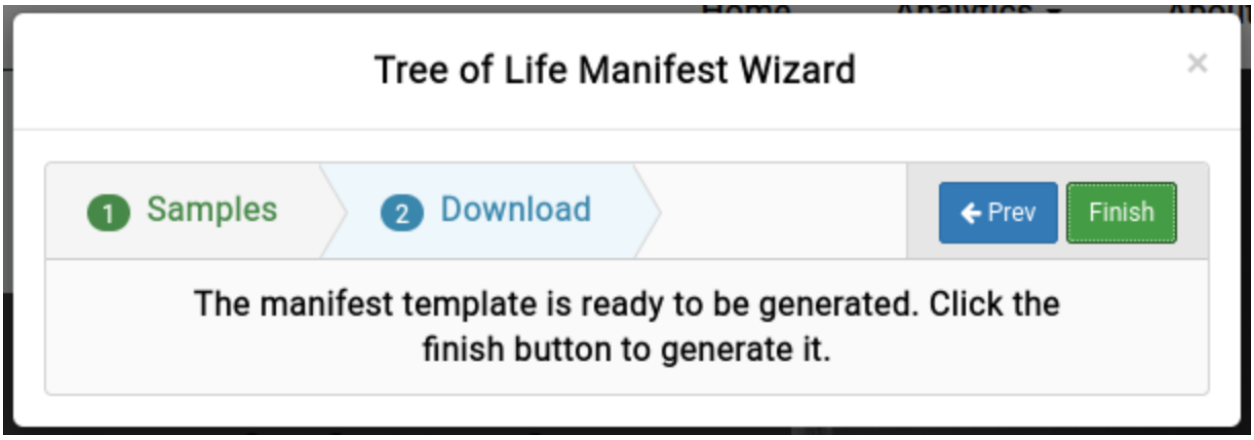
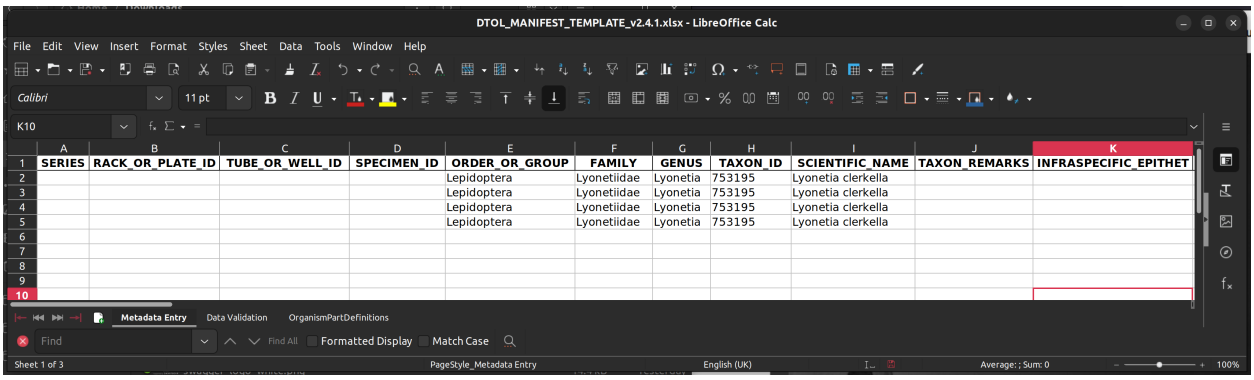


Fig. 7: **Tree of Life Manifest wizard: Download prefilled manifest by clicking the 'Finish' button**



	A	B	C	D	E	F	G	H	I	J	K
1	SERIES	RACK_OR_PLATE_ID	TUBE_OR_WELL_ID	SPECIMEN_ID	ORDER_OR_GROUP	FAMILY	GENUS	TAXON_ID	SCIENTIFIC_NAME	TAXON_REMARKS	INFRASPECIFIC_EPITHET
2					Lepidoptera	Lyonetiidae	Lyonetia	753195	Lyonetia clerkella		
3					Lepidoptera	Lyonetiidae	Lyonetia	753195	Lyonetia clerkella		
4					Lepidoptera	Lyonetiidae	Lyonetia	753195	Lyonetia clerkella		
5					Lepidoptera	Lyonetiidae	Lyonetia	753195	Lyonetia clerkella		
6											
7											
8											
9											
10											

Fig. 8: **Tree of Life Manifest wizard: Prefilled manifest**

CHAPTER

TWENTYSEVEN

TREE OF LIFE DASHBOARD

The Tree of Life (ToL) dashboard can be explored on the Tol dashboard web page .

Note: Few information will be displayed on the Tree of Life dashboard if you are not logged into COPO.

Hint:

- The Tree of Life dashboard provides an overview of COPO statistics, abbr:*ToL(Tree of Life)*[1] inspection, manifest submission and world map of Genome Acquisition Labs (GAL)s and partners.
- Click a card item on the dashboard to explore the data further.
- Click a map location on the world map to view more details about a GAL or Partner.

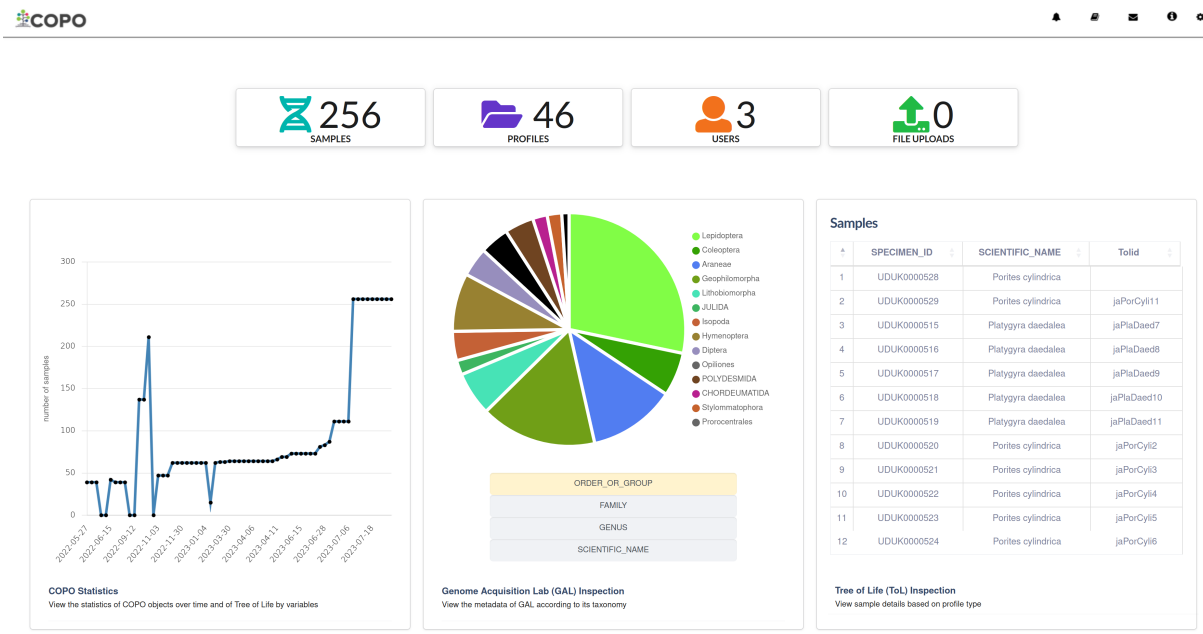


Fig. 1: **ToL Dashboard: Overview of COPO statistics and ToL inspection**

[1] See term: *Tree of Life (ToL)*

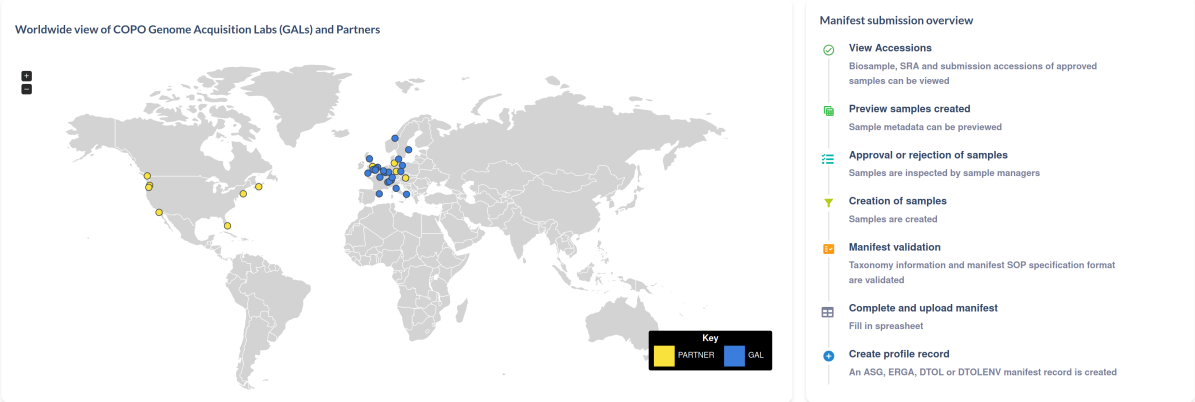


Fig. 2: **ToL Dashboard: Geographical location of all GALs and Partners and an overview of manifest submissions**

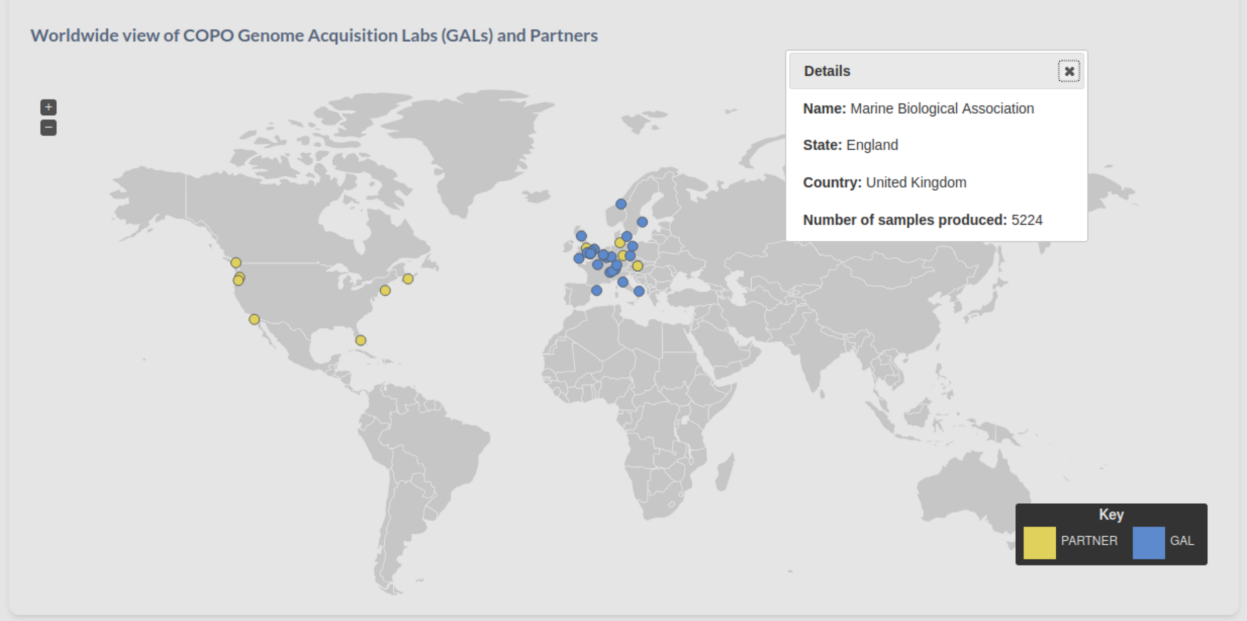


Fig. 3: **ToL Dashboard: Popup dialogue with more information about one of the GALs**
To view more details about a GAL or Partner, click one of the map locations on the world map.
A popup dialogue will be displayed information about the GAL or Partner such as name, state, country, and number of samples produced.

27.1 Tree of Life Inspection

A ToL inspection analyses the sample metadata of COPO profile records and facilitates search faceting and data discovery of the results.

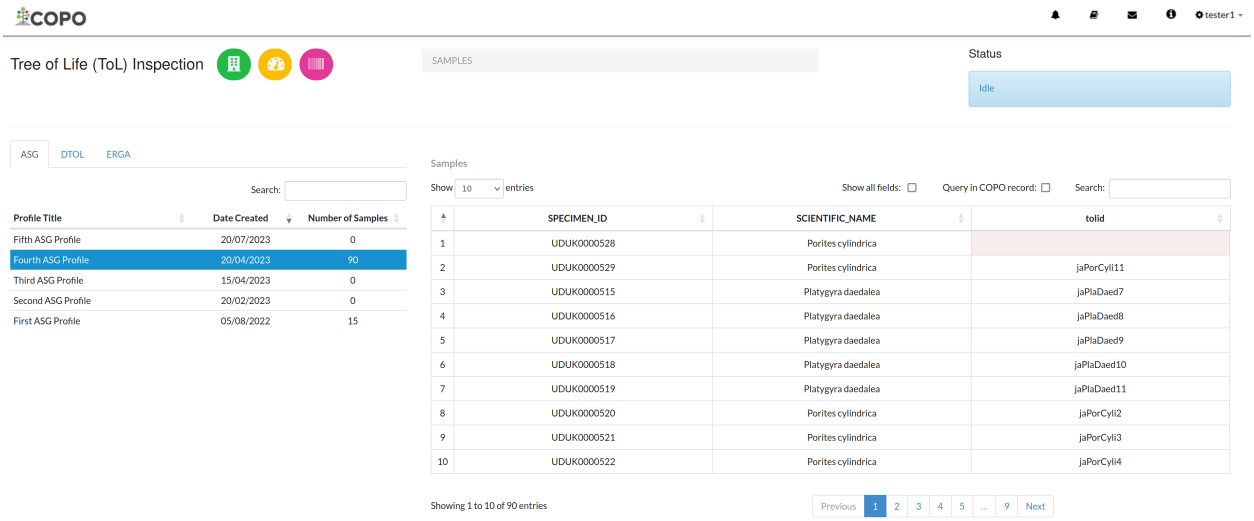


Fig. 4: **Tree of Life (ToL) Inspection web page**

27.2 Tree of Life Inspection by Genome Acquisition Lab

A ToL inspection by GAL analyses the sample metadata of COPO profile records by GALs[2] based on their taxonomy.

See also:

- *COPO statistics section*

[2] See term: *Genome Acquisition Lab (GAL)*. Also referenced as Partner.

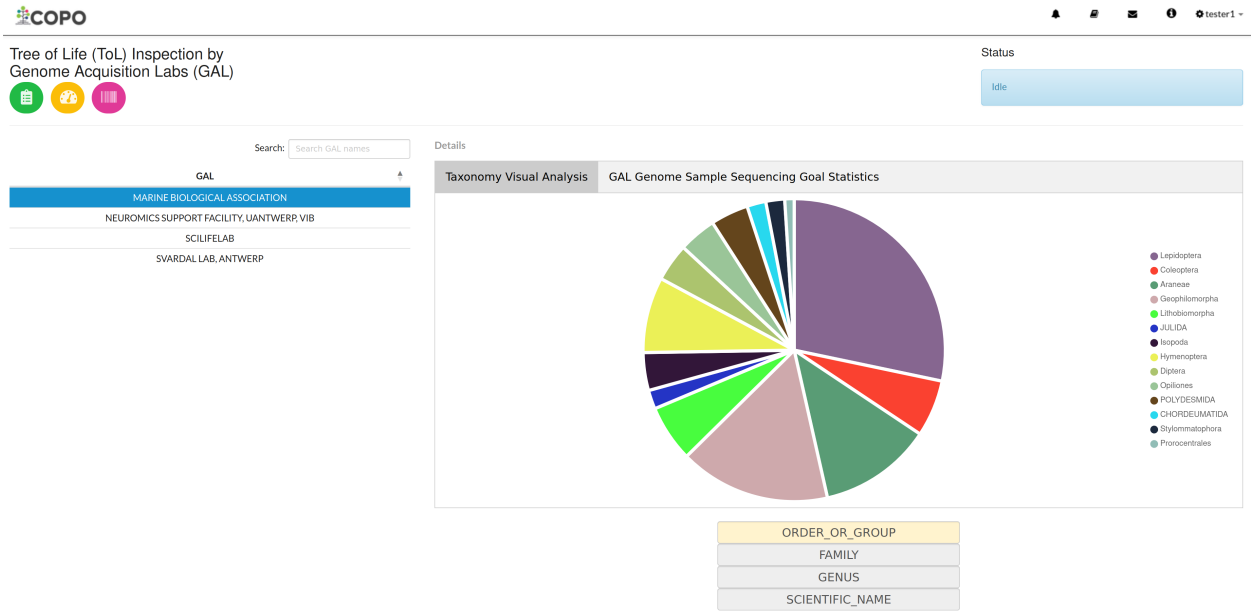


Fig. 5: **Tree of Life (ToL) Inspection by Genome Acquisition Lab (GAL) web page**

CHAPTER

TWENTYEIGHT

RETRIEVING ACCESSIONS

Approved submissions result in accessions being created. The accessions can be explored on the Accessions' dashboard , on the ENA browser[1] or NCBI Biosamples[2] (after they become public).

The types of accessions that are created are:

- **Stand-alone submissions**
  - Project accessions
  - Experiment accessions
  - Run accessions
  - Read accessions
  - Assembly accessions
  - Sequence annotation accessions
- **ToL**[Page 149, 3] **submissions**
  - Biosample accessions
  - SRA accessions
  - Submission accessions

Note:

- The button will only appear on the web page if you are granted permission to be a sample manager.

See also:

- *Stand-alone profile components*
- *ToL profile components*
- Accessions profile component

[1] See term: *ENA*.
[2] See term: *NCBI*.
[3] See term: *Tree of Life (ToL)*.

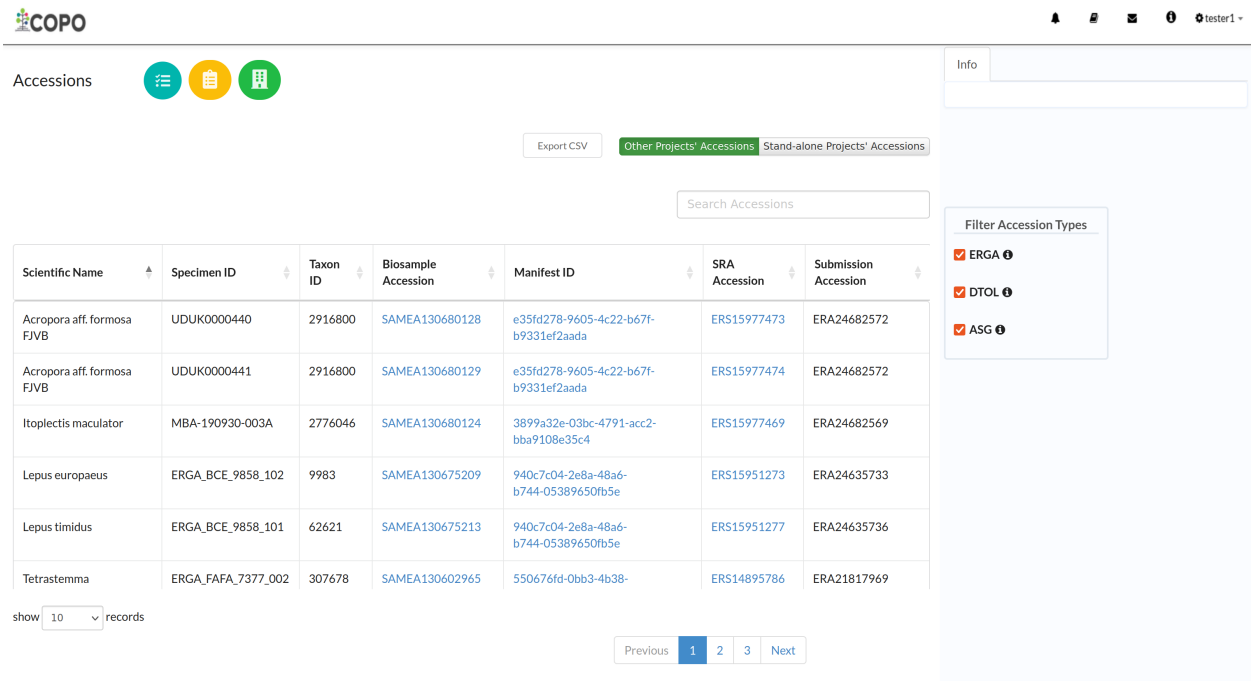


Fig. 1: **Accessions' Dashboard: Tree of Life projects' accessions**

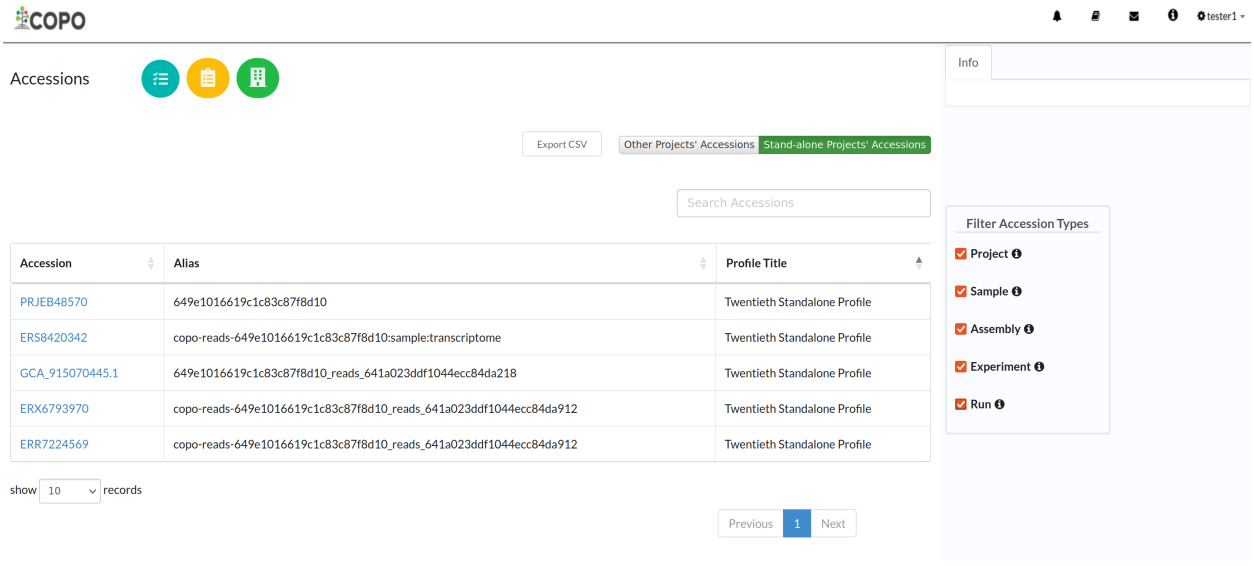


Fig. 2: **Accessions' Dashboard: Stand-alone projects' accessions**

CHAPTER

TWENTYNINE

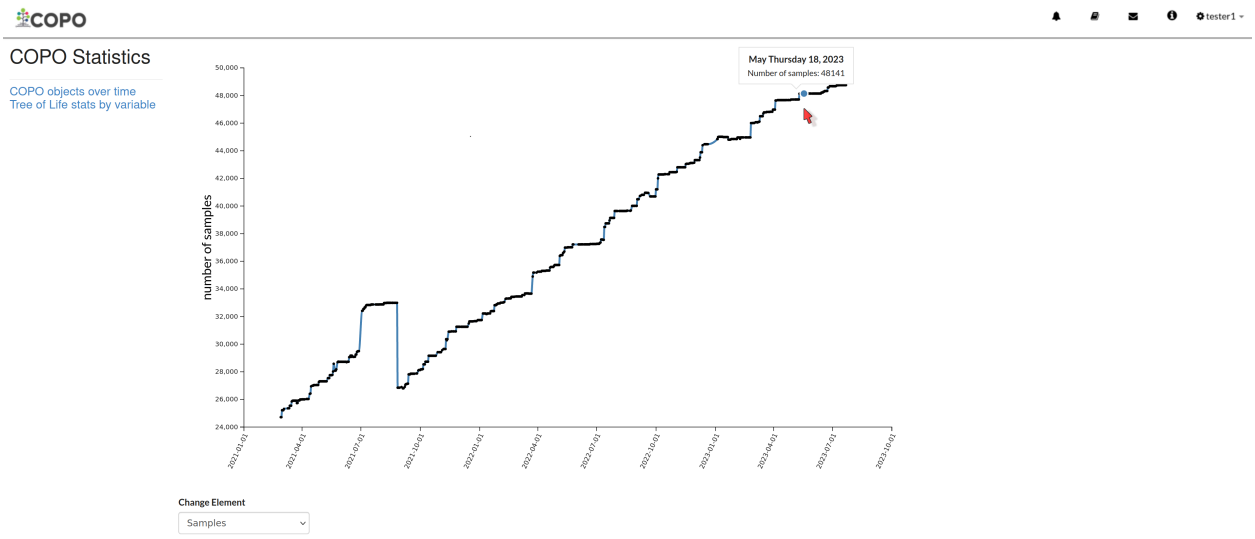
COPO STATISTICS

Statistics are generated for each COPO instance over time and by a variable. The statistics are automatically updated and can be accessed on the Statistics web page.

29.1 Statistics over Time

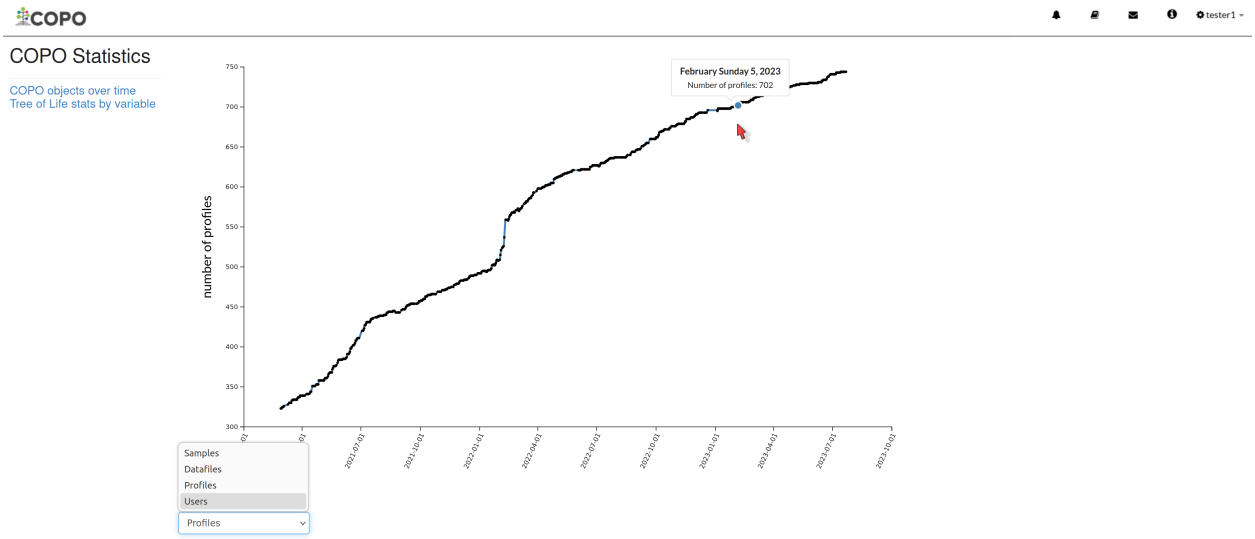
Statistics can be generated for the following research object elements over time:

- Samples
- Datafiles
- Profiles
- Users



COPO Statistics: Line graph illustrating the number of samples recorded over time

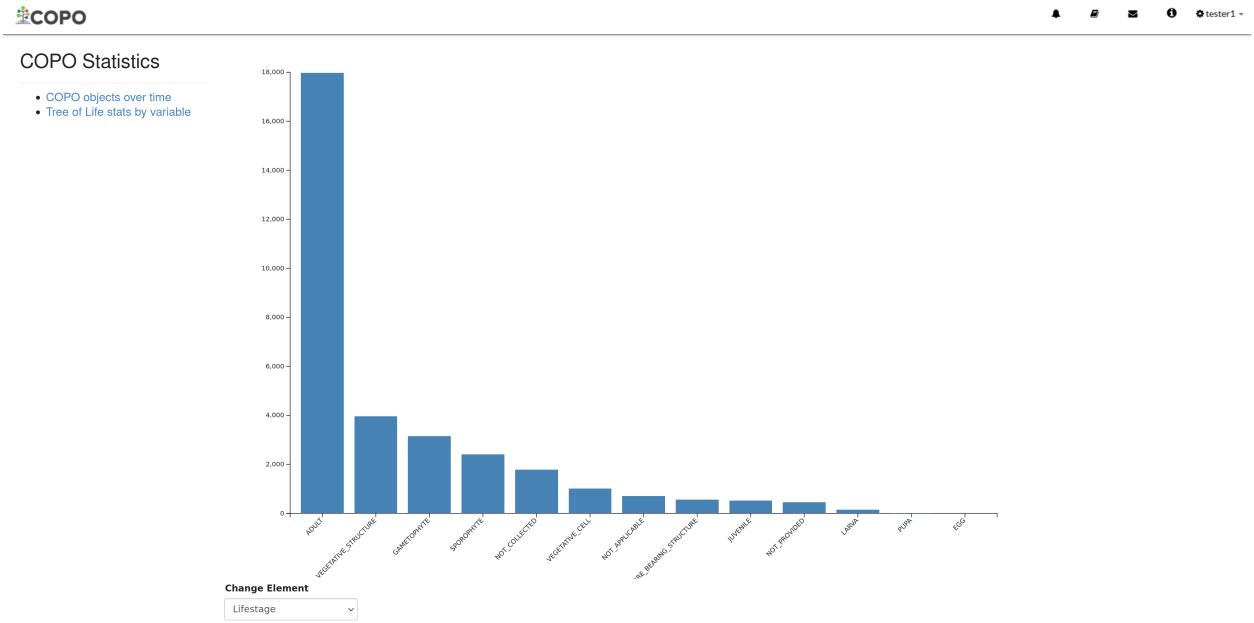
COPO Statistics: Dropdown menu of research object elements that can be queried to get statistics of the number of samples over time



29.2 Statistics by Variable

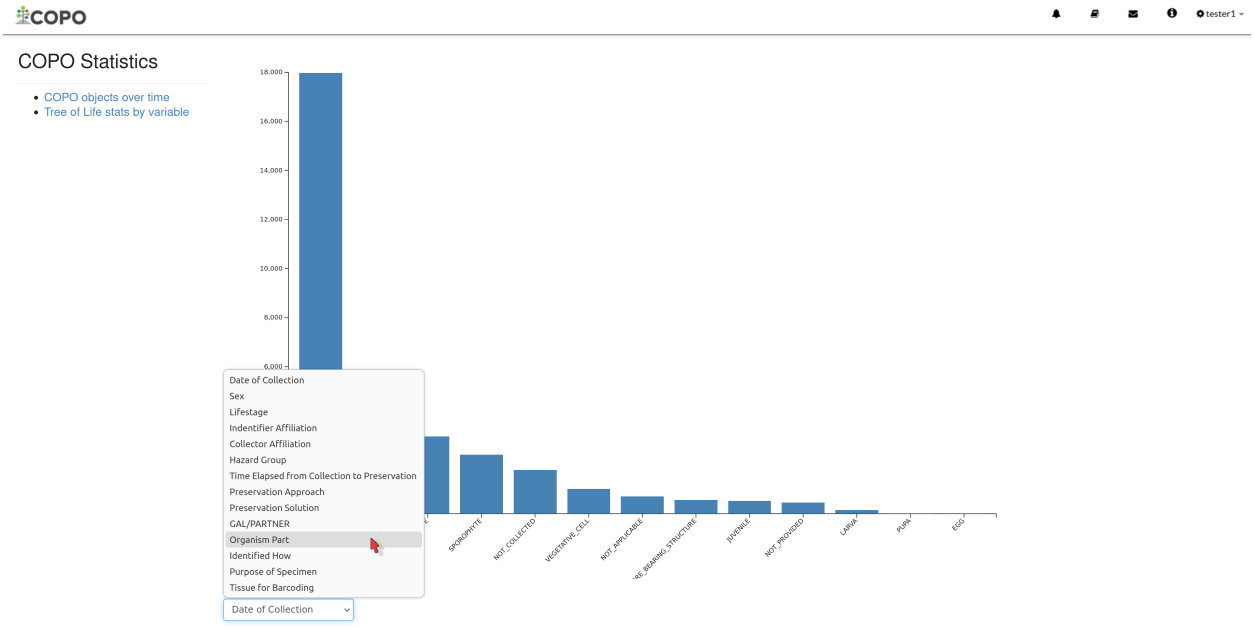
Statistics can be generated for the number of samples submitted based on the research elements or variables such as -

- Date of Collection
- Sex
- Harzard Group
- Organism Part



COPO Statistics: Bar graph illustrating total number of samples recorded for various `Lifestage` elements

COPO Statistics: Dropdown menu of elements/variables that can be queried to get statistics of the number of samples


COPO
tester1
COPO Statistics
COPO objects over time
Tree of Life stats by variable
18,000
16,000
14,000
12,000
10,000
8,000
6,000
Date of Collection
Sex
Lifestage
Indentifier Affiliation
Collector Affiliation
Hazard Group
Time Elapsed from Collection to Preservation
Preservation Approach
Preservation Solution
GAL/PARTNER
Organism Part
Identified How
Purpose of Specimen
Tissue for Barcoding
Date of Collection
SPOROPHYTE
NOT_COLLECTED
VEGETATIVE_CELL
NOT_APPLICABLE
JUVENILE
NOT_PROVIDED
LARVA
PUPA
EGG

CHAPTER

THIRTY

MANIFEST CHECKLIST

Hint: To view the entire description of a checklist, collapse the description by clicking the ▶ button below.

The manifest checklist[1] is a combination of the default European Nucleotide Archive (ENA)[2] checklist and additional checklists that are specific to projects. The manifest checklist is used to generate the manifest file which can be a Microsoft Excel (MS) spreadsheet file, Comma-separated values (CSV) file or a form that is required to be uploaded to ENA.

An overview of each manifest checklist is given below. It was adapted from ENA sample checklists and annotation checklists.

30.1 Barcoding Manifest Checklists

For mitochondrial cytochrome oxidase subunit 1 genes.

For ribosomal RNA genes from prokaryotic, nuclear or organellar DNA. All rRNAs are considered partial.

30.2 Sample Manifest Checklists

Minimum information required for the sample

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

[1] See term: *Manifest checklist*.
[2] See term: *ENA*.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Minimum information about a Micro B3 sample. A checklist for reporting metadata of marine microbial samples associated with genomics data. NOTE: Non-genomics data, i.e. oceanographic environmental data and morphology-based biodiversity data, should be submitted to the appropriate National Oceanographic Data Centre according to established reporting practices maintained by oceanographic community experts. Major National Oceanographic Data Centres from countries bordering the North-East Atlantic, and its adjacent seas: the Mediterranean, the Black Sea, the Baltic, the North Sea and the Arctic are listed at http://www.seadatanet.org/Overview/Partners.

For the Ocean Sampling Day campaign, non-genomics data shall be reported to the PANGAEA (http://www.pangaea.de/submit/).

Minimum information required for a prokaryotic pathogen sample

Minimum Data for Matching (MDM). A checklist for reporting metadata of pathogen samples for the Global Microbial Identifier (GMI) reporting system. More about GMI can be found here

Minimum information about a Tara Oceans sample. A checklist for reporting metadata of oceanic plankton samples associated with genomics data from the Tara Oceans Expedition.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Minimum information about an Influenza virus sample. A checklist for reporting metadata of Influenza virus samples associated with genomic data. This minimum metadata standard supports submission of avian, human and mammalian

surveillance data as well as serology and viruse isolate information (where available). The ENA Influenza sample checklist is based on standards in use at the Influenza Research Database.

Minimum information about a virus pathogen. A checklist for reporting metadata of virus pathogen samples associated with genomic data. This minimum metadata standard was developed by the COMPARE platform for submission of virus surveillance and outbreak data (such as Ebola) as well as virus isolate information.

Minimum Information required for reporting samples associated with genomic data, derived from carcinogen induced animal tumours. This minimum metadata standard was developed in collaboration with Duncan Odom lab for the Mouse Liver Cancer Evolution Project.

The ENA Crop sample enhanced checklist has been developed in collaboration with a number of EMBL-EBI teams to capture enriched annotation of published crop plant samples that lack sufficient reported metadata and are typically associated with systematic transcriptomic realignment-based analyses.

Minimum information about sewage samples. A checklist for reporting of sewage surveillance samples associated with sequence data from metagenomic sequencing projects. This minimum metadata standard was developed by the COMPARE platform.

ENA implementation of plant specimen contextual information associated with molecular data. The checklist has been developed in collaboration with the NCBI-GenBank and iPlant data resources under the umbrella of the Genomic Standards Consortium.

Shellfish contextual information associated with molecular data. The checklist has been developed in collaboration with EMBRIC Project partners.

GMinimum information about parasite samples. A checklist for reporting metadata of parasite samples associated with molecular data. This standard was developed by the COMPARE platform and can be used for submission of sample metadata derived from protozoan parasites (e.g. Cryptosporidium) and also multicellular eukaryotic parasites (e.g. Platyhelminthes and Nematoda).

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.

Minimum information to standardise metadata related to samples used in GMI PT (Global Microbial Identifier Proficiency Test). A checklist for reporting metadata of GMI PT samples associated with molecular data. This minimum metadata standard was developed by the COMPARE platform and can be used for submission of sample metadata derived from Campylobacter coli, Campylobacter jejuni, Listeria monocytogenes, Klebsiella pneumoniae, Salmonella enterica, Escherichia coli and Staphylococcus aureus.

Marine microalgae contextual information. The checklist has been developed in collaboration with EMBRIC Project partners and is suitable for reporting metadata related to environmental samples and those in culture collections.

A checklist for reporting metadata of human-associated pathogen samples for the COMPARE-ECDC-EFSA reporting system.

A checklist for reporting metadata of food-borne pathogen samples for the COMPARE-ECDC-EFSA reporting system.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Genomic Standards Consortium package extension for reporting of measurements and observations obtained from the environment where the sample was obtained. By choosing the environmental package, a selection of fields can be made from a relevant subsets of the GSC terms.

Minimum information to standardise metadata of binned metagenome samples. Ensures binned and MAG metagenome assembly metadata is compatible.

Minimum information required for reporting samples associated with patient-derived xenograft (PDX) models or patient samples

Minimum information required for reporting HoloFood samples. HoloFood is a 'hologenomic' approach that will improve the efficiency of food production systems by understanding the biomolecular and physiological processes affected by incorporating feed additives and novel sustainable feeds in farmed animals

Minimum information required for reporting samples associated with the Tree of Life Programme.

CHAPTER

THIRTYONE

SETTING UP COPO PROJECT LOCALLY WITH DOCKER

The central instance of COPO runs on a pool of three virtual machines. The following set up instructions are structured in a similar manner using one node. Feel free to make changes for a bigger or smaller pool.

Clone the GitHub COPO project repository.

Visual Studio Code (VSCode) is recommended for running the COPO project after having sat up the Docker environment. You can download VSCode from here for your local machine.

Note: There are a number of parameters in the command below that need to be updated or you may want to change for your local deployment. Please read through carefully.

Hint: A Python virtual environment is not required to run the COPO project application locally since the project is running via Docker containers. However, if you would like to run the project locally without Docker and within a Python virtual environment, please refer to the *COPO project setup with PyCharm documentation*.

Warning: The **ENA_SERVICE** environment variable is set to the ENA development server. All submission to this server will be deleted after 24 hours. To submit to the production ENA server remove `dev` and set `prod`.

31.1 Install Docker

Follow the official Docker installation documentation for your distribution.

Make changes to your firewall, iptables and security groups to serve a website, use Docker swarm and redis. The port number will depend on your setup and if you choose to use the default ports for each service.

31.2 Initialise Docker Swarm

Listing 1: Start a Docker swarm

```
docker swarm init --advertise-addr <the IP of the machine you want to advertise>
```

e.g. `$ docker swarm init 127.0.0.1` where `127.0.0.1` is the IP address of localhost.

This command will return a token. You need this token to make the other VMs join the swarm if you plan to use more than one one node. On the other machines run:

Listing 2: Command to make other VMs join the swarm manager

```
docker-swarm join --token <the token returned by the previous command> <the IP␣
↪advertised previously>:2377
```

Check out the Docker documentation if you want to change the default port.

Note: You may need to change the following instructions depending on the server you are deploying to. Please consider which services need to run in the backend or frontend network.

31.2.1 Add Docker Node Labels

Label the node as a web service, nginx service, mongo service, postgres service and backup service. The web service and nginx service will be deployed on the frontend network. The mongo service, postgres service and backup service will be deployed on the backend network.

Adding One Docker Node Label

Hint: `$HOSTNAME` is a bash variable that returns the hostname of the machine. You can use the hostname of the machine instead of the variable. In a Linux OS, you can run `$ echo $HOSTNAME` to get the hostname.

If you are using more than one node, you will need to label the other nodes as well.

Listing 3: Documentation for the Docker swarm command

```
docker node update \
    --label-add web-service=true \
    --label-add nginx-service=true \
    --label-add mongo-service=true \
    --label-add postgres-service=true \
    --label-add backup-service=true \
    $HOSTNAME
```

Adding More than Docker One Node Labels

If you are using more than one Docker node, you can label the other nodes as follows:

Node 1

Listing 4: Node 1 Docker label command

```
docker node update \
    --label-add web-service=true \
    --label-add nginx-service=true \
    --label-add mongo-service=false \
    --label-add postgres-service=false \
    --label-add backup-service=false \
    copo-0
```

Node 2

Listing 5: Node 2 Docker label command

```
docker node update \
        --label-add web-service=false \
    --label-add nginx-service=false \
        --label-add mongo-service=false \
        --label-add postgres-service=true \
        --label-add backup-service=true \
        copo-1
```

Node 3

Listing 6: Node 3 Docker label command

```
docker node update \
     --label-add web-service=false \
     --label-add nginx-service=false \
     --label-add mongo-service=true \
     --label-add postgres-service=false \
     --label-add backup-service=false \
     copo-2
```

Create Docker Volumes

Docker volumes are used to persist data via the plugin, **local-persist**. This ensures that the data is not lost when the containers are restarted. Volumes are created on the swarm manager.

Substitute the paths in commands before running it.

Hint: You may need to install curl before running the command below. You can install curl by running $ `sudo apt-get install curl`.

You may need to install the **local-persist** plugin to persist volumes before running the command below. You can install it by running: $ `curl -fsSL https://raw.githubusercontent.com/MatchbookLab/local-persist/master/scripts/install.sh | sudo docker`

Listing 7: Commands to create Docker volumes

```
docker volume create -d local-persist -o mountpoint=/path/to/web-data --name=web-data
docker volume create -d local-persist -o mountpoint=/path/to/static-data --name=static-
↪data
docker volume create -d local-persist -o mountpoint=/path/to/submission-data --
↪name=submission-data
docker volume create -d local-persist -o mountpoint=/path/to/logs-data --name=logs-data
docker volume create -d local-persist -o mountpoint=/path/to/mongo-data --name=mongo-
↪backup
docker volume create -d local-persist -o mountpoint=/path/to/postgres-data --
↪name=postgres-backup

docker volume create mongo-data
docker volume create postgres-data
```

Create Networks on Docker Swarm Manager

On the swarm manager create two networks - one for the frontend and one for the backend. The front-end network will be used by the web service and the nginx service while the backend network will be used by the database services.

Listing 8: Commands to create Docker networks on the swarm manager

```
docker network create -d overlay copo-frontend-network
docker network create -d overlay copo-backend-network
```

Listing 9: View networks created on Docker swarm manager

```
docker network ls
```

Create Secrets on Docker Swarm Manager

All secrets are file based. You will need to create some of the keys with third parties and choose passwords before proceeding with the COPO setup.

Listing 10: Commands to create secrets on Docker swarm manager

```
docker secret create copo_mongo_initdb_root_password copo_mongo_initdb_root_password
docker secret create copo_mongo_user_password copo_mongo_user_password
docker secret create copo_postgres_user_password copo_postgres_user_password
docker secret create copo_web_secret_key copo_web_secret_key
docker secret create copo_orcid_secret_key copo_orcid_secret_key
docker secret create copo_orcid_client_id copo_orcid_client_id
docker secret create copo_figshare_consumer_secret_key copo_figshare_consumer_secret_key
docker secret create copo_figshare_client_id_key copo_figshare_client_id_key
docker secret create copo_figshare_client_secret_key copo_figshare_client_secret_key
docker secret create copo_google_secret_key copo_google_secret_key
docker secret create copo_twitter_secret_key copo_twitter_secret_key
docker secret create copo_facebook_secret_key copo_facebook_secret_key
docker secret create copo_webin_user copo_webin_user
docker secret create copo_webin_user_password copo_webin_user_password
docker secret create copo-project.crt copo-project.crt
docker secret create copo-project.key copo-project.key
docker secret create copo_nih_api_key copo_nih_api_key
docker secret create copo_public_name_service_api_key copo_public_name_service_api_key
docker secret create copo_mail_password copo_mail_password
docker secret create copo_bioimage_path copo_bioimage_path
docker secret create ecs_secret_key ecs_secret_key
```

Listing 11: View secrets created on Docker swarm manager

```
docker secret ls
```

Build COPO Project Docker Image

Download the Dockerfile `Dockerfile_local`. for your local machine.

Place the downloaded Dockerfile in the COPO project root directory.

Alternatively, you can use the Dockerfile present in the root project directory `Dockerfile for demonstration environment`.

Note: The Dockerfile is configured to build the **local_copo_web** container image with the tag, `v1.0.1`. If you have a different tag and container name, you will need to change the Dockerfile accordingly.

If you are using a Mac OS, download the `Dockerfile_mac`.

Visit here for more information on how to build an application with a Docker image.

Listing 12: Navigate to COPO project root directory

```
cd <path-to-project-root-directory>/COPO
```

Listing 13: Build Docker image

```
docker build . -f  Dockerfile_local -t local_copo_web:v1.0.1
```

Deploy Docker Image on Docker Swarm Manager

The **redis**, **postgres** and **mongo** Docker services are created on the swarm manager. Download the `local compose file` file to create the services.

Alternatively, you can download `compose file for demonstration environment`.

Replace the `<path-to-project-root-directory>` with the absolute path to the COPO project root directory.

Note: The Docker compose file is configured to use the secrets and volumes created above. If you have used different names for the secrets and volumes, you will need to change the compose file accordingly.

If you are using a Mac OS, download the `Mac compose file`.

The following commands should be run from the root directory of the COPO project.

Warning: The **ENA_SERVICE** environment variable is set to the ENA development server. All submission to this server will be deleted after 24hours. To submit to the production ENA server remove "dev"

Listing 14: Edit Compose file to container tag e.g. local_copo_web:v1.0.1

```
nano local_copo.compose.yaml
```

Update the tag, save the file then, exit by inputting: `CTRL + O` then, `ENTER` then, `CTRL + X`

Listing 15: Command to deploy Docker image, local_copo_web:v1.0.1

```
docker stack deploy --compose-file '<path-to-file>/local_copo.compose.yaml' copo
```

Listing 16: Local Compose file for COPO project application

```
version: "3.8"
services:
  web:
    image: local_copo_web:v1.0.1
    command: >
      bash -c "code-server serve-local --disable-telemetry --auth  none --host 0.0.0.0 --
↪port 8100"
    ports:
      - "8000:8000"
      - "8100:8100"
    volumes:
      - <path-to-project-root-directory>/COPO:/copo
    environment:
      - ENVIRONMENT_TYPE=prod
      - ASPERA_PLUGIN_DIRECTORY=aspera_linux_plugin
      - SECRET_KEY_FILE=/run/secrets/copo_web_secret_key
      - MEDIA_PATH=media/
      - DEBUG=true
      - REDIS_HOST=copo_redis
      - REDIS_PORT=6379
      - WEBIN_USER_FILE=/run/secrets/copo_webin_user
      - WEBIN_USER_PASSWORD_FILE=/run/secrets/copo_webin_user_password
      - ENA_SERVICE=https://wwwdev.ebi.ac.uk/ena/submit/drop-box/submit/
      - MONGO_USER=copo_user
      - MONGO_USER_PASSWORD_FILE=/run/secrets/copo_mongo_user_password
      - MONGO_DB=copo_mongo
      - MONGO_HOST=copo_mongo
      - MONGO_PORT=27017
      - MONGO_MAX_POOL_SIZE=100
      - POSTGRES_DB=copo
      - POSTGRES_USER=copo_user
      - POSTGRES_PORT=5432
      - POSTGRES_SERVICE=copo_postgres
      - POSTGRES_PASSWORD_FILE=/run/secrets/copo_postgres_user_password
      - ORCID_SECRET_FILE=/run/secrets/copo_orcid_secret_key
      - ORCID_CLIENT_FILE=/run/secrets/copo_orcid_client_id
      - FIGSHARE_CONSUMER_SECRET_FILE=/run/secrets/copo_figshare_consumer_secret_key
      - FIGSHARE_CLIENT_ID_FILE=/run/secrets/copo_figshare_client_id_key
      - FIGSHARE_CLIENT_SECRET_FILE=/run/secrets/copo_figshare_client_secret_key
      - GOOGLE_SECRET_FILE=/run/secrets/copo_google_secret_key
```

(continues on next page)

(continued from previous page)

```
    - TWITTER_SECRET_FILE=/run/secrets/copo_twitter_secret_key
    - FACEBOOK_SECRET_FILE=/run/secrets/copo_facebook_secret_key
    - NIH_API_KEY_FILE=/run/secrets/copo_nih_api_key
    - PUBLIC_NAME_SERVICE_API_KEY_FILE=/run/secrets/copo_public_name_service_api_key
    - MAIL_PASSWORD_FILE=/run/secrets/copo_mail_password
    - MAIL_PORT=587
    - MAIL_ADDRESS=data@earlham.ac.uk
    - MAIL_SERVER=outlook.office365.com
    - MAIL_USERNAME=eidata@nbi.ac.uk
    - ALLOWED_HOSTS=
    - PUBLIC_NAME_SERVICE=https://id.tol.sanger.ac.uk/api/v2/
    - ENA_ENDPOINT_REPORT=https://wwwdev.ebi.ac.uk/ena/submit/report/samples
    - ASPERA_PATH=/root/.aspera/cli
    - BIOIMAGE_PATH_FILE=/run/secrets/copo_bioimage_path
    - BIOIMAGE_SERVER=bsaspera_w@hx-fasp-1.ebi.ac.uk
    - ECS_SECRET_KEY_FILE=/run/secrets/ecs_secret_key
    - ECS_ACCESS_KEY_ID=copo@nbi.ac.uk
    - ECS_ENDPOINT=http://ei-copo.obj-data.nbi.ac.uk
    - ENA_V2_SERVICE_SYNC=https://wwwdev.ebi.ac.uk/ena/submit/webin-v2/submit
    - ENA_V2_SERVICE_ASYNC=https://wwwdev.ebi.ac.uk/ena/submit/webin-v2/submit/queue
    - B2DROP_PERMITS=/copo/b2drop/permits
  depends_on:
    - redis
    - postgres
    - mongo
  networks:
    - copo-frontend-network
    - copo-backend-network
  deploy:
    replicas: 1
    restart_policy:
      condition: on-failure
    placement:
      constraints:
        - "node.labels.web-service==true"
    resources:
      limits:
        cpus: '4'
        memory: 7GB
      reservations:
        cpus: '2'
        memory: 2GB
    update_config:
      parallelism: 1
      delay: 10s
  secrets:
    - copo_web_secret_key
    - copo_postgres_user_password
    - copo_mongo_user_password
    - copo_google_secret_key
    - copo_facebook_secret_key
    - copo_twitter_secret_key
```

(continues on next page)

(continued from previous page)

```
    - copo_orcid_secret_key
    - copo_orcid_client_id
    - copo_figshare_client_id_key
    - copo_figshare_client_secret_key
    - copo_figshare_consumer_secret_key
    - copo_webin_user
    - copo_webin_user_password
    - copo_nih_api_key
    - copo_public_name_service_api_key
    - copo_mail_password
    - copo_bioimage_path
    - ecs_secret_key

redis:
  image: copo/copo-redis:redis-v6.2.6.18
  networks:
    - copo-frontend-network
  deploy:
    replicas: 1
    endpoint_mode: dnsrr
    restart_policy:
      condition: on-failure
    placement:
      constraints:
        - "node.labels.web-service==true"
    update_config:
      parallelism: 1
      delay: 10s
    resources:
      limits:
        cpus: "0.5"
        memory: 1GB
      reservations:
        cpus: "0.3"
        memory: 512mb

postgres:
  image: postgres:9.6
  networks:
    - copo-backend-network
  volumes:
    - postgres-data:/var/lib/postgresql/data
  secrets:
    - copo_postgres_user_password
  ports:
    - "5432:5432"
  environment:
    - POSTGRES_DB=copo
    - POSTGRES_USER=copo_user
    - POSTGRES_PASSWORD_FILE=/run/secrets/copo_postgres_user_password
  deploy:
    replicas: 1
```

(continues on next page)

(continued from previous page)

```
    endpoint_mode: dnsrr
    restart_policy:
      condition: on-failure
    placement:
      constraints:
        - "node.labels.postgres-service==true"
    update_config:
      parallelism: 1
      delay: 10s
    resources:
      limits:
        cpus: "0.5"
        memory: 1GB
      reservations:
        cpus: "0.3"
        memory: 512mb

mongo:
  image: copo/copo-mongo:20
  networks:
    - copo-backend-network
  volumes:
    - mongo-data:/data/db
  secrets:
    - copo_mongo_initdb_root_password
    - copo_mongo_user_password
  ports:
    - "27017:27017"
  environment:
    - MONGO_INITDB_ROOT_USERNAME=copo_admin
    - MONGO_INITDB_ROOT_PASSWORD_FILE=/run/secrets/copo_mongo_initdb_root_password
    - MONGO_USER=copo_user
    - MONGO_USER_PASSWORD_FILE=/run/secrets/copo_mongo_user_password
    - MONGO_DB=copo_mongo
  deploy:
    replicas: 1
    #endpoint_mode: dnsrr
    restart_policy:
      condition: on-failure
    placement:
      constraints:
        - "node.labels.mongo-service==true"
    update_config:
      parallelism: 1
      delay: 10s
    resources:
      limits:
        cpus: "0.8"
        memory: 2GB
      reservations:
        cpus: "0.4"
        memory: 512mb
```

(continues on next page)

(continued from previous page)

```
networks:
  copo-backend-network:
    external: true
  copo-frontend-network:
    external: true
secrets:
  copo_web_secret_key:
    external: true
  copo_postgres_user_password:
    external: true
  copo_mongo_user_password:
    external: true
  copo_google_secret_key:
    external: true
  copo_figshare_client_id_key:
    external: true
  copo_facebook_secret_key:
    external: true
  copo_twitter_secret_key:
    external: true
  copo_orcid_secret_key:
    external: true
  copo_orcid_client_id:
    external: true
  copo_figshare_client_secret_key:
    external: true
  copo_figshare_consumer_secret_key:
    external: true
  copo_webin_user:
    external: true
  copo_webin_user_password:
    external: true
  copo_nih_api_key:
    external: true
  copo_public_name_service_api_key:
    external: true
  copo_mail_password:
    external: true
  copo_mongo_initdb_root_password:
    external: true
  copo-project.crt:
    external: true
  copo-project.key:
    external: true
  copo_bioimage_path:
    external: true
  ecs_secret_key:
    external: true
volumes:
  web-data:
    external: true
```

(continues on next page)

(continued from previous page)

```
  static-data:
    external: true
  submission-data:
    external: true
  logs-data:
    external: true
  mongo-backup:
    external: true
  postgres-backup:
    external: true
  mongo-data:
    external: true
  postgres-data:
    external: true
  b2drop-permits:
    external: true
```

Listing 17: View services created on Docker swarm manager

```
docker service ls
docker ps
docker ps -a
```

Listing 18: Inspect created image and check if it is running

```
docker image inspect local_copo_web:v1.0.1
```

Listing 19: Start `copo_web` Docker container (if it is not started)

```
docker service scale copo_web=1
```

Listing 20: Command to stop `copo_web` Docker container

```
docker service scale copo_web=0
```

Set up PostgreSQL database

In the terminal, navigate to the root directory of the COPO project application then, run the following commands.:

Hint: Retrieve the **local_copo_web** container ID by running the $ `docker ps` command below in the root project directory of the COPO project application in the terminal for the **local_copo_web:v1.0.1** Docker image.

Listing 21: Install PostgreSQL database version 9.6 in terminal

```
docker run --name postgresql -e POSTGRES_USER=<postres-username> -e POSTGRES_PASSWORD=
↪<postres-password> -p 5432:5432 -v /data:/var/lib/postgresql/data -d postgres:9.6
```

Replace `<postgres-username>` and `<postgres-password>` with the username and password for PostgreSQL database respectively.

Listing 22: Enter the **local_copo_web** container

```
docker exec -it <local_copo_web-container-id> bash
```

Listing 23: Setup scripts to be run in the **local_copo_web** Docker container

```
python manage.py makemigrations
python manage.py makemigrations chunked_upload
python manage.py makemigrations allauth
python manage.py migrate
python manage.py setup_groups
python manage.py setup_schemas
python manage.py createcachetable
python manage.py social_accounts
```

Listing 24: Install Python requirements for the project

```
python manage.py makemigrations
python manage.py makemigrations chunked_upload
python manage.py makemigrations allauth
python manage.py migrate
python manage.py setup_groups
python manage.py setup_schemas
python manage.py createcachetable
python manage.py social_accounts
```

Listing 25: Create a Django admin/superuser

```
python3 manage.py createsuperuser
```

Enter the required details to create the Django admin/superuser. The Django admin/superuser can log into the COPO project application from the local Django admin website.

Listing 26: Exit the **local_copo_web** Docker container

```
CTRL + P
CTRL + Q
exit
```

The commands above can be accessed in the `3_db_setup.sh script`. This file is located in the **set_up_scripts** directory of the COPO project root directory.

In the following steps, we will create the PostgreSQL database for the COPO project application in the root directory of the project.

Hint: Retrieve the PostgreSQL container ID by running the command below in the root project directory of the COPO project application in the terminal for the **postgres:9.6** Docker image: $ `docker ps`

Listing 27: Enter the PostgreSQL container

```
docker exec -it <postgres-container-id> bash
```

Listing 28: Run setup scripts in the PostgreSQL Docker container

```
psql -h 'localhost' -U  $POSTGRES_USER -d 'copo' -c 'DELETE FROM socialaccount_socialapp_
↪sites'
psql -h 'localhost' -U  $POSTGRES_USER -d 'copo' -c 'DELETE FROM django_site'
psql -h 'localhost' -U  $POSTGRES_USER -d 'copo' -c 'DELETE FROM socialaccount_socialapp'
psql -h 'localhost' -U  $POSTGRES_USER -d 'copo' -c "INSERT INTO django_site (id, domain,
↪ name) VALUES (1, 'www.copo-project.org', 'www.copo-project.org')"
psql -h 'localhost' -U  $POSTGRES_USER -d 'copo' -c "INSERT INTO socialaccount_socialapp␣
↪(id, provider, name, client_id, secret, key) VALUES (1, 'orcid', 'Orcid', '$ORCID_
↪CLIENT_ID', '$ORCID_SECRET', '')"
psql -h 'localhost' -U  $POSTGRES_USER -d 'copo' -c 'INSERT INTO socialaccount_socialapp_
↪sites (id, socialapp_id, site_id) VALUES (1, 1, 1)'
```

The commands above can be accessed in the `3_db_setup.sh script`. This file is located in the **set_up_scripts** directory of the COPO project root directory.

31.2.2 Updating COPO Website Service

The COPO project is updated frequently and as such is under active development. To update your instance to a newer (or the latest) version, download the `local compose file` or the `compose file for demonstration environment` on the swarm manager or root directory of the project if you have one node.

Then, run the following commands in the terminal:

Note: The Docker tag below needs to be changed to reflect the most recent version available in DockerHub. Please check the latest version there. You can safely ignore the *feature tags as they are not stable releases. For stable releases look for `*rc`.

Hint:

- Retrieve the **copo-web** container ID by running the $ `docker ps` command below in the root project directory of the COPO project application in the terminal for the **copo/copo-web:v1.0.1** Docker image.
- To check if the web service is running, run the command below in the root project directory of the COPO project application in the terminal for the **copo/copo-web:v1.0.2** Docker image: $ `docker logs -f <container-ID-for-updated-copo-web>`
- If you update often we recommend taking care of removing old docker images regularly.

Listing 29: Edit Compose file by updating the Docker container tag on the Docker swarm manager

```
nano local_copo.compose.yaml
```

Update the Docker tag, save the file then, exit it by inputting: `CTRL + O` then, `ENTER` then, `CTRL + X`

Listing 30: Command to deploy updated Docker image: local_copo_web:v1.0.2 on the Docker swarm manager

```
docker stack deploy --compose-file '<path-to-file>/local_copo.compose.yaml' copo
```

31.2.3 Launch COPO Website

The COPO project application can be accessed locally on port 8100 via the VSCode browser extension.

Within the VSCode IDE browser, add a new configuration by following the steps below:

1. Navigate to Run -> Add Configuration
2. Edit the **launch.json** file that is created with the following file contents:

Listing 31: VSCode **launch.json** configuration file contents

```
{
        "version": "0.2.0",
        "configurations": [

            {
                "name": "Python: Django",
                "type": "python",
                "request": "launch",
                "program": "${workspaceFolder}/manage.py",
                "env" : {
                    "PYTHONPATH1" : "${workspaceFolder}/lib:${PYTHONPATH}"
                },
                "args": [
                    "runserver",
                    "0.0.0.0:8000"
                ],
                "django": true,
                "justMyCode": false
            },
            {
                "name": "Python: Django makemigration",
                "type": "python",
                "request": "launch",
                "program": "${workspaceFolder}/manage.py",
                "env" : {
                    "PYTHONPATH" : "${workspaceFolder}/lib:${PYTHONPATH}"
                },
                "args": [
                    "makemigration"
                ],
                "django": true,
                "justMyCode": false
            },
            {
                "name": "Python: Celery Workers",
                "type": "python",
```

(continues on next page)

(continued from previous page)

```
                "request": "launch",
                "module": "celery",
                "console": "integratedTerminal",
                "env" : {
                    "GEVENT_SUPPORT": "True"
                },
                "args": ["-A", "web", "worker", "-l", "debug", "-P", "gevent", "--
↪concurrency" , "10", "-Q", "celery",  "-E"],
                "justMyCode": true
            },
            {
                "name": "Python: Celery Beat",
                "type": "python",
                "request": "launch",
                "module": "celery",
                "console": "integratedTerminal",
                "args": ["-A", "web", "beat", "-l", "debug"]
             }
        ]
}
```

The COPO project application can be accessed locally on on port 80000 or on port 81000.

Note: Install required VSCode extensions for the COPO project application by following the steps below:

1. Navigate to the **Extensions** tab on the left-hand side of the VSCode IDE
2. Search for and install the following extensions:
    - Python (required)

If your local machine is restarted, you will need to start the Docker container again at startup. To do this, run the following command in the terminal: $ `docker start <container-ID-for-copo-web>`. You can retrieve the container ID by running the command below in the root project directory of the COPO project application in the terminal for the **copo/copo-web:v1.0.2** Docker image: $ `docker ps`

31.2.4 Tips

- Enable **Manage Unsafe Repositories** in **Source Control** in VSCode browser application to allow VSCode to access the COPO GitHub project repository.
- Install the following VSCode extensions:
    - GitHub Copilot
    - Prettier - Code formatter
    - Git Extension Pack

Listing 32: Set GitHub configuration in terminal

```
git config --global user.name "<GitHub-username>"
git config --global user.email "<GitHub-email-address>"
```

Listing 33: Create a tag via the terminal

```
git tag <tagname>
```

Listing 34: Push a particular tag to GitHub via the terminal

```
git push origin <tagname>
```

Listing 35: Remove an existing tag from GitHub via the terminal

```
git tag -d <tag-name>
```

Listing 36: Docker command used to list all the running Docker containers

```
docker ps
```

Listing 37: Docker command used to start, stop and restart a Docker service

```
sudo systemctl start docker
sudo systemctl stop docker
sudo systemctl restart docker
```

Listing 38: Docker command used to start, stop and restart a container respectively

```
docker start
docker stop
docker restart
```

Listing 39: Docker command used to execute a command in a running container

```
docker exec it <container-ID> bash
```

Listing 40: Docker command used to find the installed version of docker

```
docker version
```

Listing 41: Docker command used to know the details of all the running, stopped, or exited containers

```
docker ps -a
```

Listing 42: Docker command used to create a volume so that the docker container can use it to store data

```
docker volume create <volume-name>
```

CHAPTER

THIRTYTWO

SETTING UP COPO PROJECT LOCALLY WITHOUT DOCKER

The COPO project is built on the Django framework and uses MongoDB, Redis, and PostgreSQL databases. Python is the main programming language used in the project.

The open source project is hosted on GitHub.

According to your OS, please choose the appropriate link below for instructions on how to set up the project on your local machine:

32.1 Set Up COPO Project on Linux

Note: If you are using a different OS (Operating system) other than Ubuntu, you can skip this step. See *COPO project guidelines for Windows users* if Windows is your preferred OS.

32.1.1 Install Python

```
sudo apt -y install software-properties-common
sudo add-apt-repository ppa:deadsnakes/ppa
sudo apt update
sudo apt -y install python3.8
```

32.1.2 Install Python Development Tools

```
sudo apt -y install python3.8-distutils
sudo apt -y install build-essential libssl-dev libffi-dev
sudo apt -y install libxml2-dev libxslt1-dev python3-django
sudo apt update && sudo apt install -y default-jre rsync git nano libxml2-dev python␣
↪build-essential make gcc python3-dev locales python3-pip ruby-dev rubygems poppler-
↪utils
```

32.1.3 Install Required Packages

Listing 1: Install required packages and enable the databases to start on boot

```
sudo apt -y install supervisor
sudo gem install sass
sudo apt -y install redis mongodb postgresql
sudo systemctl enable mongodb
sudo systemctl enable postgresql
sudo systemctl enable redis
```

Listing 2: Install MongoDB on your local machine

```
sudo apt-key adv --keyserver hkp://keyserver.ubuntu.com:80 --recv 7F0CEB10
echo "deb http://repo.mongodb.org/apt/ubuntu "$(lsb_release -sc)"/mongodb-org/3.0␣
↪multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-org-3.0.list
sudo apt update
sudo apt -y install mongodb-org
```

32.1.4 Install Integrated Development Environment (IDE)

Any IDE (Integrated Development Environment) can be used to work on the project. However, we recommend using one of the following IDEs:

PyCharm Professional

A cross-platform IDE that provides consistent experience on the Windows, macOS, and Linux operating systems.

```
sudo snap install pycharm-professional --classic
```

OR

Visit PyCharm website to download an appropriate version for your OS or download it from the Ubuntu Software Center.

Visual Studio Code

VSCode (Visual Studio Code) is a lightweight but powerful source code editor which runs on your desktop and is available on Windows, macOS and Linux.

It comes with built-in support for JavaScript, TypeScript and Node.js and has a rich ecosystem of extensions for other languages (such as C++, C#, Java, Python, PHP, Go) and runtimes (such as .NET and Unity).

Visit VSCode website to download an appropriate version for your OS or download it from the Ubuntu Software Center

32.1.5 Configure IDE

PyCharm

Note: If using VSCode IDE, you can skip this step.

Warning: If you encounter the error, `Error: Please enable Django support for the project`, navigate to File->Settings->Languages & Frameworks->Django->Enable Django to enable Django support for the project.

If using PyCharm IDE, add a new configuration by following the steps below: #. Navigate to Add New Configuration #. Select **Django server** #. Click **Save**

Visit Run/debug configurations to learn how to create a configuration in PyCharm.

VSCode

If using VSCode IDE, add a new configuration by following the steps below: #. Navigate to Run -> Add Configuration #. Edit the **launch.json** file that is created with the following file contents:

Listing 3: VSCode **launch.json** configuration file contents

```
{
        "version": "0.2.0",
        "configurations": [

            {
                "name": "Python: Django",
                "type": "python",
                "request": "launch",
                "program": "${workspaceFolder}/manage.py",
                "env" : {
                    "PYTHONPATH1" : "${workspaceFolder}/lib:${PYTHONPATH}"
                },
                "args": [
                    "runserver",
                    "0.0.0.0:8000"
                ],
                "django": true,
                "justMyCode": false
            },
            {
                "name": "Python: Django makemigration",
                "type": "python",
                "request": "launch",
                "program": "${workspaceFolder}/manage.py",
                "env" : {
                    "PYTHONPATH" : "${workspaceFolder}/lib:${PYTHONPATH}"
                },
                "args": [
```

(continues on next page)

(continued from previous page)

```
                    "makemigration"
                ],
                "django": true,
                "justMyCode": false
            },
            {
                "name": "Python: Celery Workers",
                "type": "python",
                "request": "launch",
                "module": "celery",
                "console": "integratedTerminal",
                "env" : {
                    "GEVENT_SUPPORT": "True"
                },
                "args": ["-A", "web", "worker", "-l", "debug", "-P", "gevent", "--
↪concurrency" , "10", "-Q", "celery",  "-E"],
                "justMyCode": true
            },
            {
                "name": "Python: Celery Beat",
                "type": "python",
                "request": "launch",
                "module": "celery",
                "console": "integratedTerminal",
                "args": ["-A", "web", "beat", "-l", "debug"]
             }
        ]
}
```

32.1.6 Declare Environment Variables

Set secrets in the environment variables in your **.bashrc** file. Activate **.bashrc** file by running the command: `source .bashrc`

If using PyCharm IDE, declare the environment variables in the following places: * Edit Configurations > Environment variables * File->Settings->Build, Execution, Deployment->Console->Python console * File->Settings->Build, Execution, Deployment->Console->Django console * File->Settings->Languages & Frameworks->Django->Enable Django

32.1.7 Set up Python virtual environment

Run the following commands within the project directory to set up a Python virtual environment for the project

Alternatively, you can create a virtual environment for the project with the command: `python3 -m venv venv`

Listing 4: Install packages with pip3 and Python3

```
sudo apt install -y python3-virtualenv
source venv/bin/activate
pip3 install -r requirements/base.txt
pip3 install celery
```

(continues on next page)

(continued from previous page)

```
python manage.py migrate && python manage.py social_accounts && python manage.py setup_
↪groups && python manage.py setup_schemas && python manage.py createcachetable &&␣
↪supervisord -c celery.conf && supervisorctl -c celery.conf start all
python manage.py createsuperuser
```

Warning: Here are solutions to possible errors that you might encounter after having ran the above commands:

- **Possible error #1**: TypeError: 'Collection' object is not callable. If you meant to call the 'authenticate' method on a 'Database' object it is failing because no such method exists.

  Solution #1: Run the command: `$ pip3 install pymongo==3.11.4` **Reason**: pymongo versions >4 is incompatible with the COPO project

- **Possible error #2**: Issues related to **importlib**, **rdlib**, **lxml**, **numpy**, **pandas** and **cffi** Python packages

  Solution #2: #. In the `requirements/base.txt` file located in the project directory, change the version numbers for the following packages so that all packages can be compatible with each other

  - `asgiref==3.7.2`
  - `cffi==1.15.1`
  - `importlib-metadata==1.6.1`
  - `lxml==4.9.3`
  - `numpy==1.20.0`
  - `pandas==1.2.2`

  1. Change line 12 in **wizard_helper.py** file from `pandas.io.json import json_normalize` to `from pandas import json_normalize`
  2. Run the command: `pip3 install -r requirements/base.txt` again to install the packages

- **Possible error #2**: text index required for $text query

  Solution #3: Create an index in the SampleCollection in mongoDB: `db.SampleCollection.createIndex( { "$**": "text" } )`

32.1.8 Set Up Mongo Database in COPO Project

Note: Replace the username and password for MongoDB with your own username and password

Run the **4_mongo_setup.sh** MongoDB set up script to configure the MongoDB database on your machine

Set up MongoDB in Studio3T

Note: Replace the <username> and <password> for MongoDB with your MongoDB username and password By default, the username for MongoDB is `copo_user`. The hostname,``localhost``,, can also be substituted with the IP address of `127.0.0.1`

Create a new MongoDB connection manually with the following details: * **Connection Name**: `copo_mongo` * **Connection Type**: `Standalone Connection` * **Hostname**: `localhost` * **Port**: `27017` * **Authentication**: Username and Password * **Username**: `copo_user` * **Password**: <password> * **Authentication Database**: `admin`

OR

MongoDB URI: `mongodb://copo_user:password@localhost:27017/copo_mongo?retryWrites=true&connectTimeoutMS=10000&authSource=admin&authMechanism=SCRAM-SHA-1`

32.1.9 Set Up PostgreSQL Database in COPO Project

Note: Replace the username and password of PostgreSQL with your own username and password

Run the **3_db_setup.sh** PostgreSQL set up script to configure the PostgreSQL database on your machine

32.1.10 Launch COPO Website

Listing 5: Launch COPO website using the following command

```
python manage.py runserver
```

Alternatively, click on the **Run** button (i.e the green play button) on the top-right corner of the PyCharm IDE to launch the website.

32.2 Set Up COPO Project on Windows

32.2.1 Option 1: Using Ubuntu Terminal

Set up Ubuntu terminal on Windows by enabling the **Windows Subsystem for Linux** through the following steps.

Navigate to Control Panel–>Programs–> Turn Windows Features On Or Off # Enable the **Windows Subsystem for Linux** option in the list # Click the **OK** button # Click **Restart now** when you're prompted to restart your computer # Open Microsoft Store (MS) from the **Start** menu, and search for **Linux** in the store or open this link to find the Ubuntu app on MS store: # Click **Get the apps** under the **Linux on Windows?** banner. # Search and click Ubuntu Linux distribution to get "Get" or "Install" it # Launch Ubuntu app

Note: The first time that the Linux environment is launched, one is prompted to enter a UNIX username and password. These don't have to match your Windows username and password, but will be used within the Linux environment.

Hint: In the Ubuntu terminal (which can be found in the Microsoft Store or from the **Start** menu or by searching for **Ubuntu** using the search bar): * To copy text: highlight desired text then, `right-click` * To paste text: `right-click`

Warning:

- **Error #1**: 0x800701bc WSL 2 requires an update to its kernel component. For information please visit https://aka.ms/wsl2kernel **Solution**: Go to Manual installation steps for older versions of WSL | Microsoft Docs to download WSL 2Set WSL 2 as default version: wsl –set-default-version 2
- **Error #2**: WslRegisterDistribution failed with error: 0x80370102. Error: 0x80370102 The virtual machine could not be started because a required feature is not installed. **Solution**: * Run Windows PowerShell under Administrator rights (Right-click on the Windows 10 Start button and select "Windows PowerShell (Admin)" * Copy-paste the following command: * Enable-WindowsOptionalFeature -Online -FeatureName VirtualMachinePlatform * Restart the PC or laptop

32.2.2 Option 2: Using Windows Terminal

Note:

- Commands are entered in the command prompt
- Cloning a GitHub repository enables the branches to show and gives access to committing, pushing etc to the remote repository
- Repeat steps 1 to 19 under Ubuntu section in setting up COPO project on Ubuntu

Hint:

- To open the command prompt in Windows, open the **Start menu** and search for `cmd.`. Press **Enter** or click the result to open a command window or `right-click` the option to run it as an administrator
- To view all environment variables in the terminal:C:Users<username> SET

1. Download and install Python:
2. Download PyCharm: Python IDE for Professional Developers by JetBrains
3. Install Django: >``py -m pip3 install Django``
4. Download PyCharm IDE for Professional Edition:
5. Setup PyCharm Professional with a JetBrains account and a school email
6. Add/Authorise GitHub account to PyCharm
7. Clone the COPO project GitHub repository instead of downloading it as a `.zip` file from the GitHub repository
8. Open the cloned COPO project in PyCharm
9. In PyCharm, add a new configuration by navigating to - Add Configurations->Add new->Django Server
10. Set up or copy and paste environment variables in the following four places in PyCharm: * Edit Configurations * File->Settings->Build, Execution, Deployment->Console->Python console * File->Settings->Build, Execution, Deployment->Console->Django console * File->Settings->Languages & Frameworks->Django->Enable Django

11. Set system environment variables * On the Windows taskbar, `right-click` the Windows icon and select **System**. * In the **Settings** window, under R**elated Settings**, click **Advanced system settings**. * On the **Advanced** tab, click **Environment Variables** * Click **New* to create a new environment variable * Click **Edit** to modify an existing environment variable * After creating or modifying the environment variable, click **Apply** and then, **OK** to have the change take effect

OR

Set environment variables in the terminal * To set environment variables in the terminal replace "export" with "set"

e.g. Replace `> export ENVIRONMENT_TYPE="dev"` with `> set ENVIRONMENT_TYPE="dev"`

CHAPTER

THIRTYTHREE

BROKERING DATA IN COPO WITH FAIR DATA PRINCIPLES

Findable, Accessible, Interoperable and Reusable (FAIR) data enables data consumers (people or machines) to find, aggregate and analyse data that would otherwise be private or invisible, building on the existing web of linked data.

In COPO, the FAIR principles are applied as follows:

33.1 Findable

33.1.1 Using DataTables' search engine

COPO uses *DataTables* to display tabular data. The data are indexed in the DataTables' *smart* search engine and the search box is accessible within each table of data; it is located at the top right of the table. The search engine is updated automatically when metadata is added or updated.

The search box provides the following abilities:

- Match words out of order
- Partial word matching
- Preserved text

See here for more information about searching for data in DataTables.

33.1.2 Using COPO API

Metadata can also be found via the *COPO API*.

33.2 Accessible

The required data can be retrieved freely with unrestricted access or via authentication and authorisation mechanisms, where necessary.

33.3 Interoperable

Data are integrated and available on ENA and NCBI. Data can be exported in csv, json or ro-crate format using the COPO API.

33.4 Reusable

Metadata are well-described with a plurality of accurate and relevant attributes. They are published to public repositories and can be accessed with the COPO API via a clear and accessible data usage.

33.5 FAIR Webinars

Webinar: COPO: Extending the frontiers of "FAIR" Data in Agriculture as part of RDA/IGAD webinar series

Date: February 12th July, 2020.

Host: Research Data Alliance/Improving Global Agricultural Data (RDA/IGAD)

Featured:

- Dr. Anthony Etuk , former Research Software Engineer of the COPO team
- Dr. Felix Shaw , Research Software Engineer of the COPO team

https://youtu.be/33Ozdpdfyh0

See also:

Visit FAIR website for more information about FAIR principles.

CHAPTER

THIRTYFOUR

LICENCE

COPO is licensed under the Massachusetts Institute of Technology (MIT) licence. See below for details.

Copyright (c) 2017-2024, Collaborative OPen Omics Project (COPO) Project Development and Documentation

Permission is hereby granted, free of charge, to any person obtaining a copy of this documentation and associated software files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

CHAPTER

THIRTYFIVE

CONTRIBUTOR COVENANT CODE OF CONDUCT

35.1 Our Pledge

We as members, contributors, and leaders pledge to make participation in our community a harassment-free experience for everyone, regardless of age, body size, visible or invisible disability, ethnicity, sex characteristics, gender identity and expression, level of experience, education, socio-economic status, nationality, personal appearance, race, religion, or sexual identity and orientation.

We pledge to act and interact in ways that contribute to an open, welcoming, diverse, inclusive, and healthy community.

35.2 Our Standards

Examples of behaviour that contributes to a positive environment for our community include:

- Demonstrating empathy and kindness toward other people
- Being respectful of differing opinions, viewpoints, and experiences
- Giving and gracefully accepting constructive feedback
- Accepting responsibility and apologising to those affected by our mistakes, and learning from the experience
- Focusing on what is best not just for us as individuals, but for the overall community

Examples of unacceptable behaviour include:

- The use of sexualised language or imagery, and sexual attention or advances of any kind
- Trolling, insulting or derogatory comments, and personal or political attacks
- Public or private harassment
- Publishing others' private information, such as a physical or email address, without their explicit permission
- Other conduct which could reasonably be considered inappropriate in a professional setting

35.3 Enforcement Responsibilities

Community leaders are responsible for clarifying and enforcing our standards of acceptable behaviour and will take appropriate and fair corrective action in response to any behaviour that they deem inappropriate, threatening, offensive, or harmful.

Community leaders have the right and responsibility to remove, edit, or reject comments, commits, code, wiki edits, issues, and other contributions that are not aligned to this Code of Conduct, and will communicate reasons for moderation decisions when appropriate.

35.4 Scope

This Code of Conduct applies within all community spaces, and also applies when an individual is officially representing the community in public spaces. Examples of representing our community include using an official e-mail address, posting via an official social media account, or acting as an appointed representative at an online or offline event.

35.5 Enforcement

Instances of abusive, harassing, or otherwise unacceptable behaviour may be reported to the community leaders responsible for enforcement at . All complaints will be reviewed and investigated promptly and fairly.

All community leaders are obligated to respect the privacy and security of the reporter of any incident.

35.6 Enforcement Guidelines

Community leaders will follow these Community Impact Guidelines in determining the consequences for any action they deem in violation of this Code of Conduct:

35.6.1 1. Correction

Community Impact: Use of inappropriate language or other behaviour deemed unprofessional or unwelcome in the community.

Consequence: A private, written warning from community leaders, providing clarity around the nature of the violation and an explanation of why the behaviour was inappropriate. A public apology may be requested.

35.6.2 2. Warning

Community Impact: A violation through a single incident or series of actions.

Consequence: A warning with consequences for continued behaviour. No interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, for a specified period of time. This includes avoiding interactions in community spaces as well as external channels like social media. Violating these terms may lead to a temporary or permanent ban.

35.6.3 3. Temporary Ban

Community Impact: A serious violation of community standards, including sustained inappropriate behaviour.

Consequence: A temporary ban from any sort of interaction or public communication with the community for a specified period of time. No public or private interaction with the people involved, including unsolicited interaction with those enforcing the Code of Conduct, is allowed during this period. Violating these terms may lead to a permanent ban.

35.6.4 4. Permanent Ban

Community Impact: Demonstrating a pattern of violation of community standards, including sustained inappropriate behaviour, harassment of an individual, or aggression toward or disparagement of classes of individuals.

Consequence: A permanent ban from any sort of public interaction within the community.

35.7 Attribution

This Code of Conduct is adapted from the Contributor Covenant, version 2.0, is available at https://www.contributor-covenant.org/version/2/0/code_of_conduct.html.

Community Impact Guidelines were inspired by Mozilla's code of conduct enforcement ladder.

For answers to common questions about this code of conduct, see the FAQ or Translations.

Last reviewed on: 08-01-2024

CHAPTER

THIRTYSIX

PRIVACY NOTICE

This Privacy Notice explains what data is collected by the **Collaborative OPen Omics (COPO)** website service, its purposes, how the data is processed and how COPO keeps it secure.

Please view our *Terms of Use* regarding the rules and requirements for the use of the service.

36.1 Consent

Processing your data is necessary for our legitimate interest of allowing the operation and functioning of COPO. By using our website, you hereby consent to our Privacy Notice and agree to its terms.

36.1.1 Data Collected

COPO adopts appropriate data collection, validation, storage and processing practices as well as security measures to protect against unauthorised access, alteration, disclosure or destruction of your personal information, Orcid credentials and metadata provided on our service.

Information collected:

- Internet Protocol (IP) address
- Operating system
- Browser
- First name, last name and username after you sign into the service with your Orcid credentials
- Email address (after you input it in the dialog shown after a first-time access to the service)
- Date and time when a profile record and manifests are created or updated
- Metadata submitted. Metadata is the essential information about when, where, how and why data has been collected. It can relate to research elements like samples, assemblies, sequencing annotations, barcoding, experiments, reads or runs
- If you contact us via email, we may receive additional information about you such as the contents of the message and/or attachments you may send us and any other information you may choose to provide

Usage of information collected:

- To utilise Google Fonts, an interactive web directory via Cascading Style Sheets (CSS), which is embedded in the service
- To provide the user access to the service
- To answer support request sent by the user
- To monitor errors and exceptions that may occur during metadata submission
- To register your email address with your COPO user account and to submit your email address to public repositories after manifest submissions
- To send email notifications when submitted samples have been accepted or rejected by a sample manager

36.2 Data Access

- Authorised COPO team developers
- Public metadata repositories that COPO relies on to provide the service. They are:
  - BioImage Archive (BIA)
  - ENA[1]
  - NCBI[2]

36.3 Data Retention

- Any personal data (name and email address) and metadata obtained from you will be retained as long as the service is live, even if you stop using the service.
- COPO obscures your email address and omits your name from the COPO Application Programming Interface (API) method results after a manifest submission. However, your email address will be submitted to public repositories and will be visible on public records as a metadata submitter.
- Access to the web and security logs containing personal data are restricted to the relevant COPO team developers

36.4 Cookies

This service uses "cookies" to store information about the web pages that the visitor has accessed or visited.

The information is used to log any errors or exceptions that may occur and user sessions.

[1] See term: *ENA*.
[2] See term: *NCBI*.

36.5 Questions

Any questions or comments concerning this Privacy Notice can be addressed to us by

Published on: 28-10-2023

CHAPTER

THIRTYSEVEN

TERMS OF USE

- The **Collaborative OPen Omics (COPO)** website service promotes *Findable, Accessible, Interoperable and Reusable (FAIR)* metadata brokerage through the submission of research objects (raw or processed data, reads, samples, images etc.) which are made public to the research community and public repositories like -
  - BioImage Archive (BIA)
  - ENA[1]
  - NCBI[2]
- This service uses cookies to record information about your website session to enable the website to function well. You can control your use of cookies from your web browser but if you choose not to accept cookies from the COPO website service, you will not be able to take full advantage of all of the website's features.
- This service also uses Google Fonts, a website font service that is owned by Google LLC or by Google Ireland Limited, to provide an interactive web directory via Cascading Style Sheets (CSS) throughout the service.

  The Google server needs to collect your Internet Protocol (IP) address in order to send you the Google Font files from the Google servers when you access the service. Through the use of Google Fonts, the website performance is optimised and is displayed more beautiful.

  Even though our service self-host Google Fonts on some web pages in order to eliminate violating General Data Protection Regulation (GDPR), there are other web pages in our service that do not. By using the service, you consent for the service to use your IP address so that it can load the Google Fonts from the Google servers.
- COPO will make all reasonable effort to maintain continuity of its service and provide adequate warning of any changes or discontinuities. However, COPO accepts no responsibility for the consequences of any temporary or permanent discontinuity of the service.
- If you email or submit offensive, inappropriate or objectionable content on our website or to us, or engage in any disruptive behaviour on our service, we may use your personal information from our security logs to stop such behaviour. We may also use your personal information to inform relevant third parties about the content and your behaviour where we reasonably believe that you are or may be in breach of any applicable laws.
- COPO imposes no additional restrictions on the use of the submitted data available on our service after they have been provided by the metadata submitter other than those provided by the metadata submitters like embargoed metadata.

  However, there are restrictions imposed on the metadata submitter on the metadata after the metadata have been uploaded and submitted on the COPO service.

  Some of these metadata submitter restrictions include but are not limited to the following:
- Profile records cannot be deleted after they have associated research objects like samples, reads, files, sequencing annotations, accessions, barcoding manifests etc..

[1] See term: *ENA*.
[2] See term: *NCBI*.

- Samples cannot be deleted after they have been submitted; they can only be updated. More details about what field values can be updated can be found in the *Updating Samples* section in our documentation.
- Please see our documentation on the various profile components for *Stand-alone Profile* and *Tree of Life Profile* for more information concerning the restrictions and possibilities of the various research objects.
- All scientific metadata will be made publicly available after data submission and we may store it permanently.
- The metadata submitted and the COPO Application Programming Interface (API) methods are generated in part from the data contributed by the community who remain the data submitters.
- COPO does not guarantee the accuracy of any provided metadata for any purpose. We, however, make all reasonable effort to ensure that the metadata submitted are validated according to the Standard Operating Procedure (SOP)[3] for each manifest type.

  Please see our documentation regarding the *SOP requirements* for each manifest type for more information.
- The metadata submitted may be subject to embargoed rights or permits rights. It is the responsibility of users of this service to ensure that their exploitation of the data does not infringe any of the rights of such parties.
- We reserve the right to update these terms of use at any time. We will attempt to give reasonable notice of any changes by placing a notice on our website or by sending an email to any email address you may have provided to us, but you may wish to check the terms of use each time you use the website.

  The date of the most recent revision will appear at the bottom of this web page, the 'Terms of Use' web page. If you do not agree to these changes, please do not continue to use our service.
- Any feedback provided to COPO about its service will be treated as non-confidential unless the individual or organisation providing the feedback states otherwise.
- Any questions or comments concerning these terms of use can be addressed to us by . More details about the data that we collect from you when you browse the website, make manifest submissions and how we process the data are detailed in our *Privacy Notice*.

Last revised: 28-10-2023

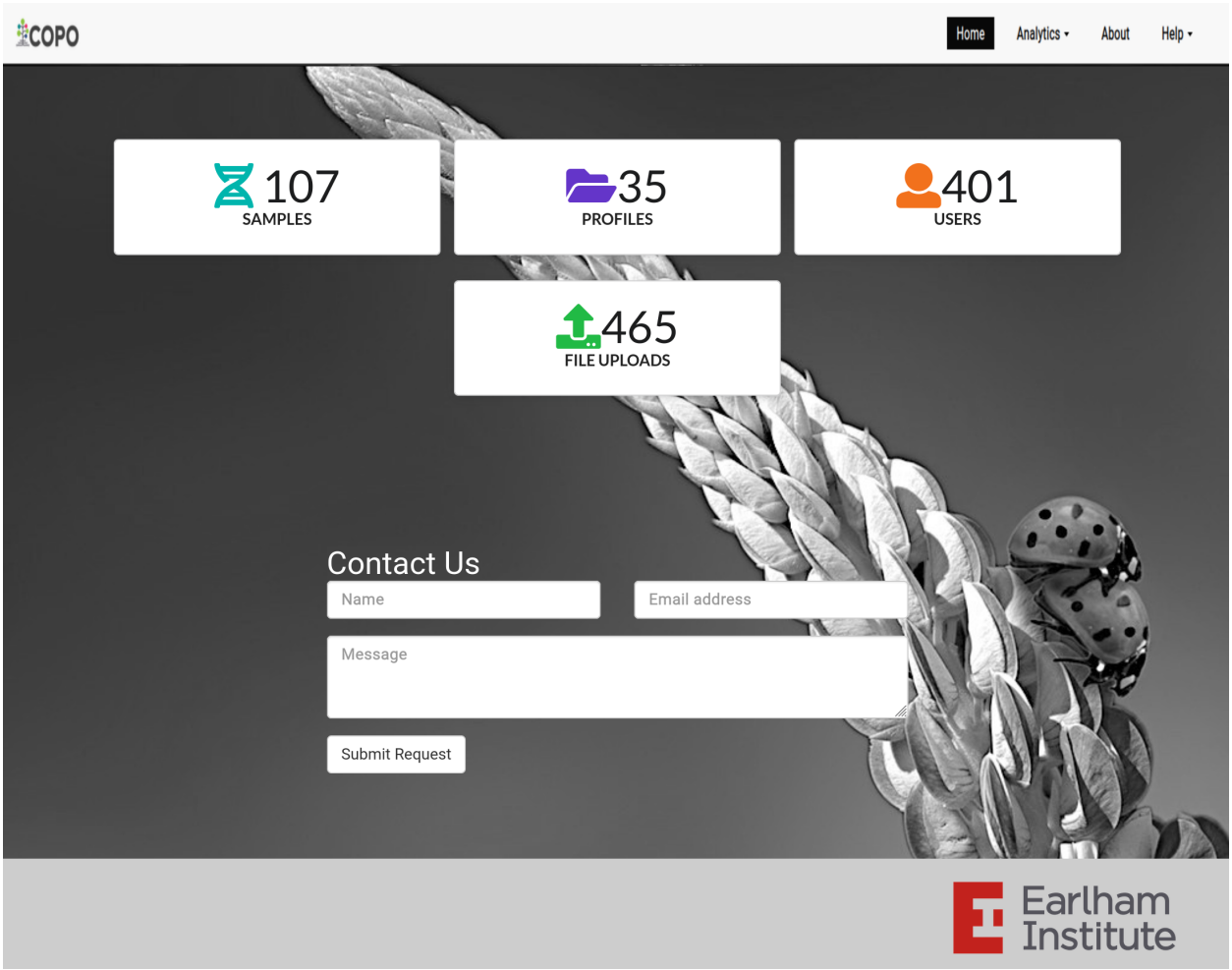
[3] See term: *SOP*.

CHAPTER

THIRTYEIGHT

CONTACT COPO

Send an email to the COPO team at , indicating your enquiry or request. The COPO team will respond as soon as possible.



Alternatively, you can input your enquiry or request in the Contact form on the COPO homepage

Alternatively, click the envelope icon located in the top-right section of the top navigation bar to be redirected to the contact form on the COPO homepage (if you are already logged into COPO)

CHAPTER

THIRTYNINE

FREQUENTLY ASKED QUESTIONS

Hint: To view the entire answer to a question, collapse the answer by clicking the ▶ button below.

39.1 Dashboard

39.1.1 How can I view accessions after I have submitted samples, reads, or experiments in COPO?

- Click the button.
- The accessions dashboard will be displayed.
- Alternatively, navigate to the Accessions' dashboard.

39.1.2 Is there a way to analyse metadata submissions?

1. **Tree of Life dashboard**
    - Alternatively, click the button.
2. **Tree of Life inspection web page**
    - Alternatively, click the button.
3. **Tree of Life inspection by Genome Acquisition Lab web page**
    - Alternatively, click the button.
4. Statistics web page

39.2 Manifest Update

39.2.1 How can I update values for samples that I have submitted in COPO?

Note:

- The manifest must be reuploaded in the same profile that the samples were submitted in.

- The desired value(s) will be updated once the field value is not a compliance field[1].
- See the *Updating Samples* section for information about which field values can be updated.

39.3 Permits

39.3.1 Can I view or download permits that I have uploaded in COPO?

Yes, permits can be retrieved and downloaded by selecting the desired sample record(s) on the **Samples** web page

Then, clicking the

button on the web page.

39.3.2 Why am I unable to upload permit one after the other?

Warning:

- If you have more than one permit file to upload, they **must** be uploaded at the same time i.e. after you have clicked the

button, navigate to the directory where the permits are stored and `CTRL + click` all of the permits so that all the permits are highlighted and uploaded at the same time.

- All permit files have to be selected/opened from the directory and uploaded together not one after the other.

39.4 Profiles

39.4.1 How can I be added to a profile group?

39.4.2 How can I create a profile on COPO?

- View the following video to see how to create a profile.

https://youtu.be/7xiVTNw6pPc

[1] See term: *Compliance field*

39.4.3 How can I upload/submit research objects to a profile owned by another user or how can I create a group or how can I share my profile with others?

See *Sharing Profiles* section for more information.

39.4.4 How can I add a subproject to a profile on COPO?

- In the **Add Profile** form, choose the desired subproject(s) from the list of associated projects as shown below.
- See the *Secondary Projects* section for information about the available subprojects.

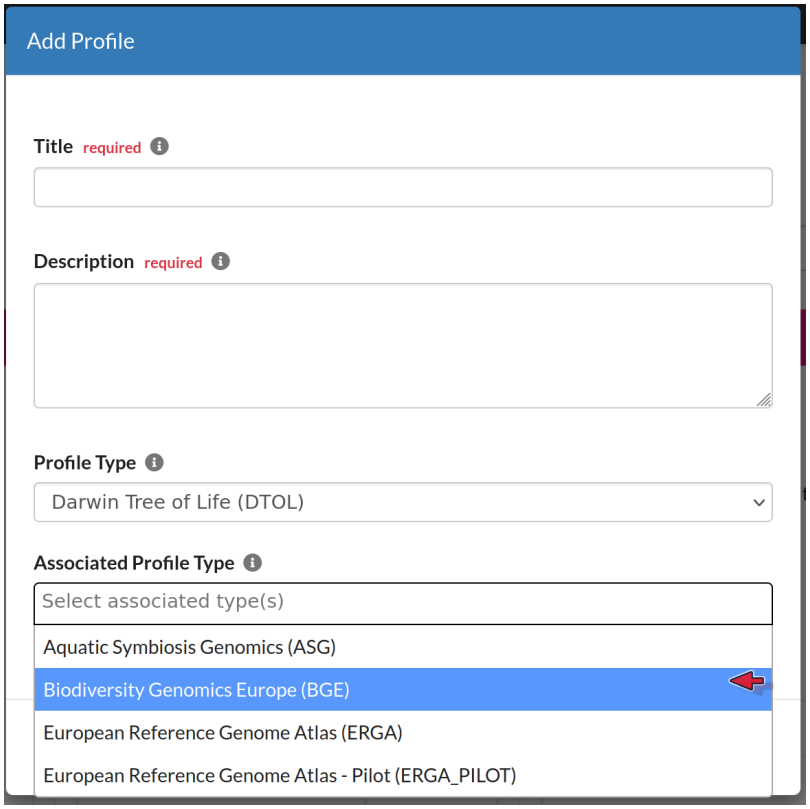


Fig. 1: **Add Profile form: Selecting Associated Profile Type dropdown menu**

39.4.5 How can I add a subproject or secondary project to a primary project in ENA/Biosamples?

- Contact the with the request providing the project accession of the child/subproject and the project accession of the parent/primary project.

39.4.6 How can I edit or delete a profile that I have created in COPO?

- Click the ⋮ icon that is associated with the desired profile.
- The option to edit or delete a profile record will be displayed once clicked.
- The web page will refresh after the task has been completed successfully.

39.4.7 How many profiles can I have in COPO?

- For instance, you can create a profile to represent work done as part of a grant, subproject within a project or a PhD (Doctorate of Philosophy) project.

39.4.8 How can I view more profiles that I have created in COPO?

- More profiles that you have created will be loaded.

39.4.9 How can I view more information about a profile that I have created in COPO?

Note:

- The View more... button will only be associated with a profile if the profile has at least one of the following information.

- Click the View more... button associated with the profile.
- After the button is clicked, a popup dialog will show at least one of the following information if it is available:
  - **Release Status** (if applicable)
  - **Release Date** (if applicable)
  - **Associated Profile Type(s)**
  - **Sequencing Centre**

39.4.10 How can I navigate to the top of the web page after having loaded several work profiles?

- Alternatively, click the button which automatically navigates to the top of the web page.

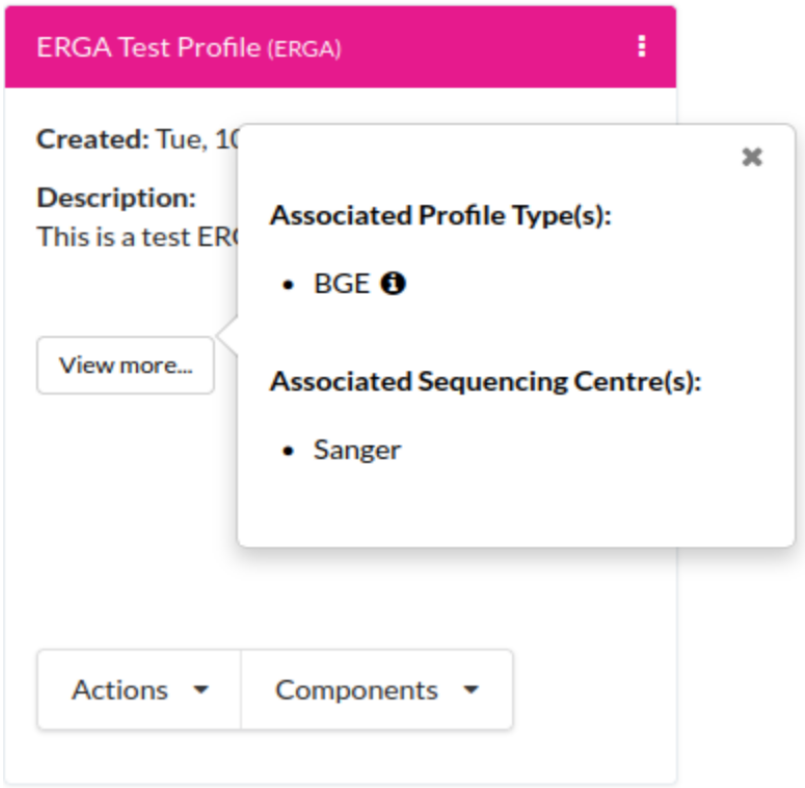


Fig. 2: **Profile: View more information popup dialog**

39.4.11 How can I create components for a profile?

1. Navigate to the work profile web page
2. Click the button
3. Fill in then, save the form that is displayed
4. Click the

button associated with the created profile record to view a dropdown list of tasks that can be performed for that profile
5. Complete the action/task
6. Click the

button associated with the profile record to view the component of the action that was performed

39.5 Samples

39.5.1 Can I delete samples that have submitted?

No, samples cannot be deleted after the manifest have been submitted.

39.5.2 Can I retrieve samples or the manifest that have been submitted?

Yes, see the *How to Download Submitted Sample Manifest* section for more information.

39.5.3 How can I view images that have been uploaded?

See the *View Submitted Images* section for more information.

39.5.4 How can I download permits that have been uploaded?

See the *Download Submitted Permits* section for more information.

39.5.5 What are the formats that I download samples in?

The following are the formats that samples can be downloaded in:

- Microsoft Excel Spreadsheet format (`.xlsx`)

  See the *Downloading manifest in spreadsheet format* section for more information.

- Comma-separated values (csv) format (`.csv`)

  On the **Samples** web page, click the

button to download a manifest in csv format.

  See How to access Samples web page section for guidance.

39.6 Sample Managers

39.6.1 How can I be assigned as a sample manager?

- Make a request to the indicating the type of profile group that you would like to be assigned as a sample manager.
- The permission will be granted after the request has been approved.

39.6.2 How can I know if I have been assigned as a sample manager?

- The button will be displayed on the web page.
- The **accept/reject samples** web page will be displayed once the button is clicked.
- Alternatively, if you can navigate to the Accept/Reject Samples' web page with an **Unauthorisation** error web page being displayed then, you are a sample manager.

39.6.3 How can I accept or reject samples that users have submitted?

Note: See *Accessing the Accept or Reject Samples web page* section for guidelines on accessing the **Accept or Reject Samples** web page.

See *Accepting or Rejecting Samples* section for more information.

39.6.4 Can I download permits that users have uploaded in COPO?

Yes, on the **Accept or Reject Samples** web page, permits can be downloaded by selecting the desired sample record(s) then, clicking the

button on the web page.

See *Downloading permits* section for more information.

39.6.5 Can I view images that users have uploaded in COPO?

Yes, on the **Accept or Reject Samples** web page, images can be viewed by selecting the desired sample record(s) then, clicking the

button on the web page.

See *Viewing images* section for more information.

39.6.6 If I belong to more than one sample manager manifest group, how can I view or accept samples that belong to them?

Note:

- The manifest dropdown menu will only be displayed on the **Accept or Reject samples** web page if you as a sample manager, belongs to more than one sample manager manifest group.
- If the *dtol* sample manager group dropdown menu option is selected, both Aquatic Symbiosis Genomics (ASG) profiles and Darwin Tree of Life (DToL) profiles will be displayed in the **All profiles** tab and/or **Profiles for My Sequencing Centre** the **Accept or Reject samples** web page will be displayed (if any exists).

- Choose desired sample manager group from the manifest group dropdown menu.
- Click the button to accept or reject samples.

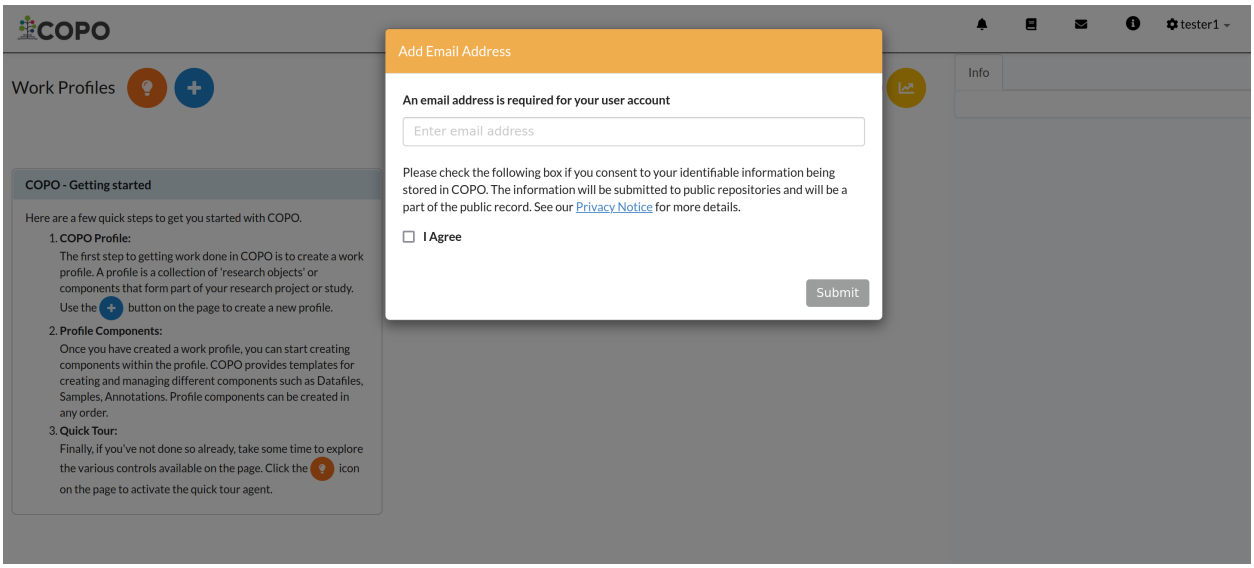


Fig. 3: **Accept or Reject samples: Email address prompt shown when a user logs into COPO for the first time**

See *Viewing images* section for more information.

CHAPTER

FORTY

GLOSSARY

API
A software interface that allows two or more applications to communicate with each other.

ASG
A project that aims to provide the genomic foundations needed by scientists to answer key questions about the ecology and evolution of symbiosis in marine and freshwater species, where at least one partner is a microbe.

Assembly
Any sequence records from coding or non-coding regions to full assembled chromosomes.

BGE
A project that undertakes a comprehensive application of genomic science to biodiversity research will drive fundamental advances in conservation science and policy.

Biocuration
The extraction of unstructured biological data from manifests into a structured, computable form.

Compliance field
A compliance field in COPO is a mandatory field that cannot be updated after a manifest has been uploaded or submitted before or after the samples have been accepted or rejected by a sample manager.

COPO (Collaborative OPen Omics)
COPO is a web-based tool for creating and managing metadata for research objects.

COPO profile
Also known as work profile. A collection of 'research objects' or components that contain data generated on a biological research project or study.

There are two general types of profiles in COPO: *Stand-alone* profiles and ToL profiles. ASG, DToL and ERGA are the ToL (primary) projects brokered through COPO.

CRUD (Create, Read, Update and Delete) operations
Comprises posting data (creating and/or updating data), reading data (e.g. making queries) and deleting data.

DataTables
A jQuery library plug-in that displays tabular data. They are used in COPO to display a list of research objects. See DataTables for more information.

DB (Database)
A database is an organised collection of data, generally stored and accessed electronically from a computer system.

DNA (Deoxyribonucleic acide)
The molecule inside cells that contains the genetic information responsible for the development and function of an organism.

Docker
A set of platform as a service (PaaS) products that use OS-level virtualisation to deliver software in packages called containers. Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels.

Docker Engine is the software that oversees the hosting of the containers.

Dockerfile
A text-based file with no file extension that contains a script of instructions. Docker uses this script to build a container image. See Dockerfile reference

DToL
Aims to sequence the genomes of 70,000 species of eukaryotic organisms in Britain and Ireland.

It is one of the ToL projects brokered through COPO. The project is a collaboration between biodiversity, genomics and analysis partners that is transforming the manner by which biology, conservation and biotechnology are conducted.

DToL may sometimes be referred to as *DTOL*.

EBI (European Bioinformatics Institute)
EBI is a UK (United Kingdom) government-funded public repository for biological data that provides free access to biomedical and genomic information.

EI (Earlham Institute)
The Earlham Institute is a hub of life science research, training, and innovation focused on understanding the natural world through the lens of genomics. EI supports several projects including COPO project.

EI is is one of four Norwich BioScience Institutes (NBI) international centres that is based in the Norwich Research Park in Eastern England. It is also one of eight institutes that receive strategic funding from the Biotechnology and Biological Sciences Research Council (BBSRC) , part of UKRI (United Kingdom Research and Innovation), as well as support from other research funders.

The other NBI (Norwich BioScience Institutes) centres are John Innes Centre (JIC), Sainsbury Laboratory (TSL) and Quadram Institute Bioscience (QIB).

EMBL (European Molecular Biology Laboratory)
EMBL is a EU (European Union) intergovernmental organisation that performs basic research in molecular biology and provides services to the scientific community in its member states.

ENA
ENA is a repository for nucleotide sequence data that provides annotated DNA and RNA sequences . It also provides free and unrestricted access stores complementary information such as experimental procedures that details sequence assemblies and other metadata related to sequencing projects.

ENA is part of the International Nucleotide Sequence Database Collaboration (INSDC), which comprises the DNA DataBank of Japan (DDBJ), the European Molecular Biology Laboratory (EMBL), and GenBank at the National Center for Biotechnology Information (NCBI).

ERGA
A project that revolves around a pan-European scientific response to current threats to biodiversity by studying reference genomes which provide the most complete insight into the genetic basis that forms each species and represent a powerful resource in understanding how biodiversity functions.

FAIR
The ability to find, access, interoperate and reuse data with no or minimal human intervention.

GAL (Genome Acquisition Lab)
Partners or companies that perform genome sequencing.

Genome
A complete set of genetic material stored in long molecules of DNA in living organisms such as virus, oak tree or an elephant.

Genomics
The study of all or a substantial portion of the genes of an organism as a dynamic system, over time, to determine how those genes interact and influence biological pathways, networks, and physiology.

HTTP (Hypertext Transfer Protocol)
A protocol that allows communication between different systems. Similar to HTTPS (Hypertext Transfer Protocol Secure).

See also: HTTPS protocol

IP (Internet Protocol)
A protocol that allows communication between different systems.

MacOS
A series of proprietary graphical operating systems developed and marketed by Apple Inc. since 2001.

Manifest
A csv (comma-separated values) file or Microsoft (MS) Excel spreadsheet that contains metadata regarding a research object.

The manifest is used by scientists to upload metadata into COPO.

Manifest checklist
A list of fields that are required to be filled in for a sample to be considered valid.

Manifest ID
A unique identifier assigned to each manifest record in COPO.

Metadata
In-depth and controlled contextual information about when, where, how and why data has been collected like geographical location, time of collection, tube or well identification and specimen identification. Metadata can relate to a research elements such as samples, assembles, annotations or experiments.

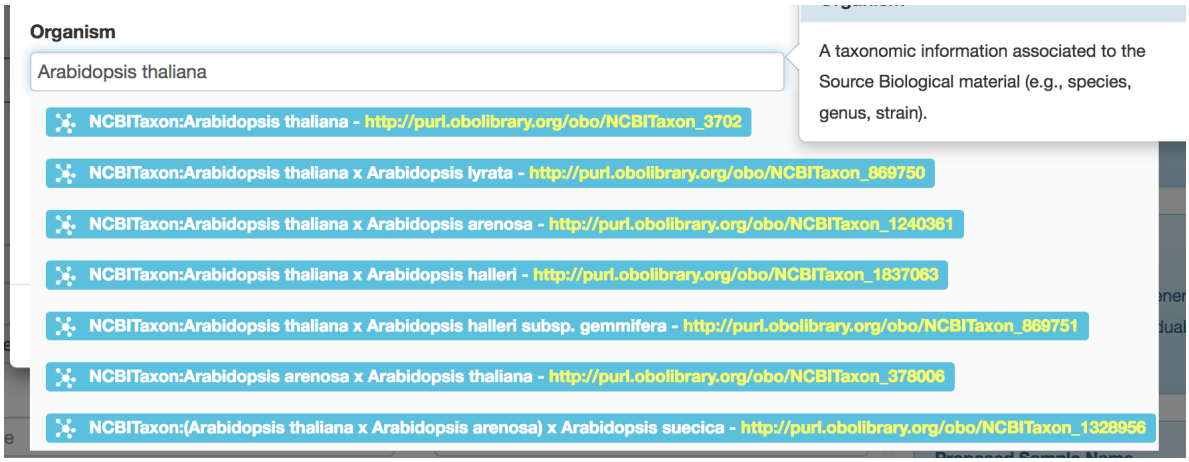
In life sciences, metadata facilitates *biocuration* which revolves around the structuring of datasets in a way that allows automated search, query and retrieval.

MIT (Massachusetts Institute of Technology) licence
A permissive free software license from the Massachusetts Institute of Technology that has limited restriction on the reuse of software.

MongoDB
A document-oriented database program that uses JSON-like documents with optional schemas.

NCBI (National Center for Biotechnology Information)
NCBI is a US (United States) government-funded public repository for biological data that provides free access to biomedical and genomic information.

Ontology field
An ontology field in COPO is an auto-complete control, that maps user input to semantically-enriched values, which are explicit formal specifications of terms and their relationship to other terms in a domain.



Profile component
: Also known as *research object*. It forms part of a research project or study.

  Templates for creating or describing research objects can be found *here*.

Profile Types legend
: This describes the types of the profiles that have been created. It is located at the right of the * *Work Profiles** web page.

PyCharm
: A Python IDE (Integrated Development Environment) that provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems and supports web development with Django.

Read the Docs
: A documentation hosting service based around Sphinx. COPO documentation is hosted on Read the Docs.

Reads
: A research object that holds raw read files and sequencing methods that refer to the DNA sequence from a small section of DNA.

  It can be associated with one or more files, assemblies and sequence annotations.

Research
: Systematic investigation, including research development, testing, and evaluation, designed to develop or contribute to generalisable knowledge.

REST
: Relies on a stateless, client-server and cacheable communications *HTTP* request protocol. In COPO, it is used to communicate with the COPO API to perform *CRUD operations* using HTTP requests.

RNA-seq (Ribonucleic acid sequencing)
: Analysis based on next-generation sequencing (NGS) data has recently become the de facto standard for the analysis of gene expression at the level of the whole transcriptome.

RO-Crate (Research Object Crate)
: RO-Crate is based on schema.org annotations in JSON-LD, and aims to make best-practice in formal metadata description accessible and practical for use in a wider variety of situations, from an individual researcher working with a folder of data, to large data-intensive computational research environments.

  See more information about RO-Crate here.

Sample
: Also known as *biosample*. A research object that represents biological samples collected and sequenced in real life.

Sample checklist
The checklist of metadata that the sample was registered with.

Sample manager
A sample manager is a person who is responsible for accepting or rejecting samples in a research project.

This person can also upload manifest on behalf of sample submitters.

Sample submitter
A sample submitter is a person who submits or uploads samples to a research project.

Sample submitters may also be referred to as manifest providers or manifest submitters.

Sequence annotation
A research object that is used to describe the process of marking specific features in a DNA, RNA or protein sequence with descriptive information about structure or function.

It can be associated with one or more files, reads and assemblies.

Singular stage
In datafile description, a **singular stage** is a stage of the description wizard in which all the files in the description bundle are constrained (by the system) to share the same metadata.

SOP (Standard Operating Procedure)
A manual compiled by various profile groups to help scientists fill in a manifest correctly.

See the *SOPs <fill-blank-manifests>* section for more information.

Specimen
Also known as biospecimen. It is a piece or portion of tissue, urine or other biologically derived material used for diagnosis and analysis.

SRA (Sequence Read Archive) accession
A unique identifier assigned to a sample by the Sequence Read Archive (SRA) database. It usually starts with 'ERS' followed by a number.

Studio3T
Studio3T, formerly known as Robo3T, is a GUI (Graphical User Interface) for MongoDB.

Visit Studio3T to download an appropriate version for your OS.

ToL
A worldwide collaborative effort of biologists and nature enthusiasts to provide information about biodiversity, the characteristics of different groups of organisms and their evolutionary history (phylogeny).

Ubuntu
A Linux (Linux) distribution based on Debian and composed mostly of free and open-source software.

URI (Uniform Resource Identifier)
A string of characters that unambiguously identifies a particular resource.

See also: Uniform Resource Identifier

VSCode
VSCode is a lightweight but powerful source code editor which runs on your desktop and is available on Windows, macOS and Linux.

INDEX

A

API, 207
ASG, 207
Assembly, 207

B

BGE, 207
Biocuration, 207

C

Compliance field, 207
COPO, 207
COPO profile, 207
CRUD operations, 207

D

DataTables, 207
DB, 207
DNA, 207
Docker, 208
Dockerfile, 208
DToL, 208

E

EBI, 208
EI, 208
EMBL, 208
ENA, 208
ERGA, 208

F

FAIR, 208

G

GAL, 208
Genome, 209
Genomics, 209

H

HTTP, 209

I

IP, 209

M

MacOS, 209
Manifest, 209
Manifest checklist, 209
Manifest ID, 209
Metadata, 209
MIT licence, 209
MongoDB, 209

N

NCBI, 209

O

Ontology field, 209

P

Profile component, 210
Profile Types legend, 210
PyCharm, 210

R

Read the Docs, 210
Reads, 210
Research, 210
REST, 210
RNA-seq, 210
RO-Crate, 210

S

Sample, 210
Sample checklist, 211
Sample manager, 211
Sample submitter, 211
Sequence annotation, 211
Singular stage, 211
SOP, 211
Specimen, 211
SRA accession, 211

Studio3T, 211

T

ToL, 211

U

Ubuntu, 211
URI, 211

V

VSCode, 211