The EBRAINS Knowledge Graph

Introductory Tutorial

Laura Morel, Alix Bonard, Peyman Najafi

NeuroPSI Paris-Saclay Institute of Neuroscience, CNRS, Université Paris-Saclay, France.

12 March 2025 – Accelerate Your Neuroscience Research with EBRAINS: EBRAINS tutorials and users day, Heidelberg (Germany)

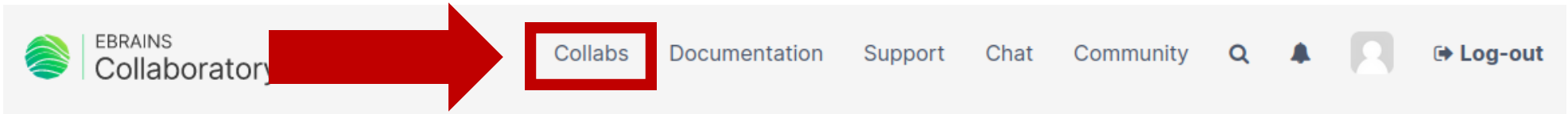
Introduction

Setup: creating a collab in the Collaboratory

- Available here: https://wiki.ebrains.eu
- The EBRAINS Collaboratory allows users to coordinate and work together in an online environment to publish work, document methodology, share data
- Collabs are workspaces in the Collaboratory

Setup: creating a collab in the Collaboratory

- https://wiki.ebrains.eu

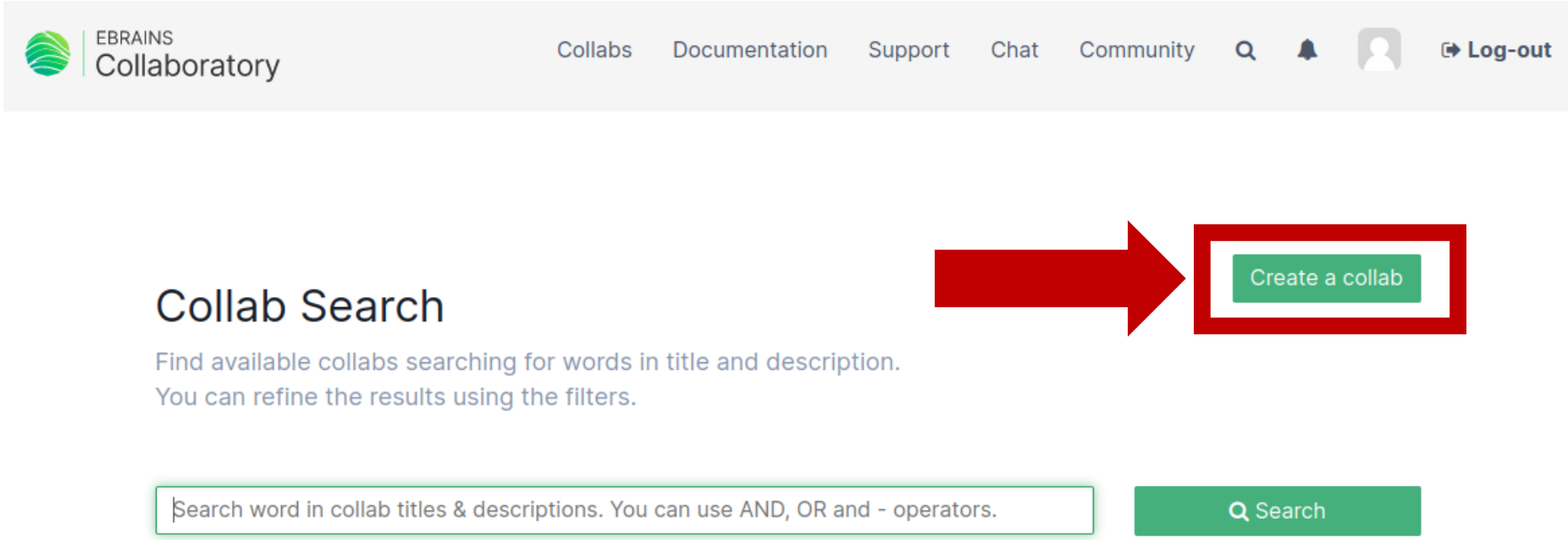


Collaborate.
Create reproducible science.
Discover EBRAINS services at work.
From anywhere.

Getting started

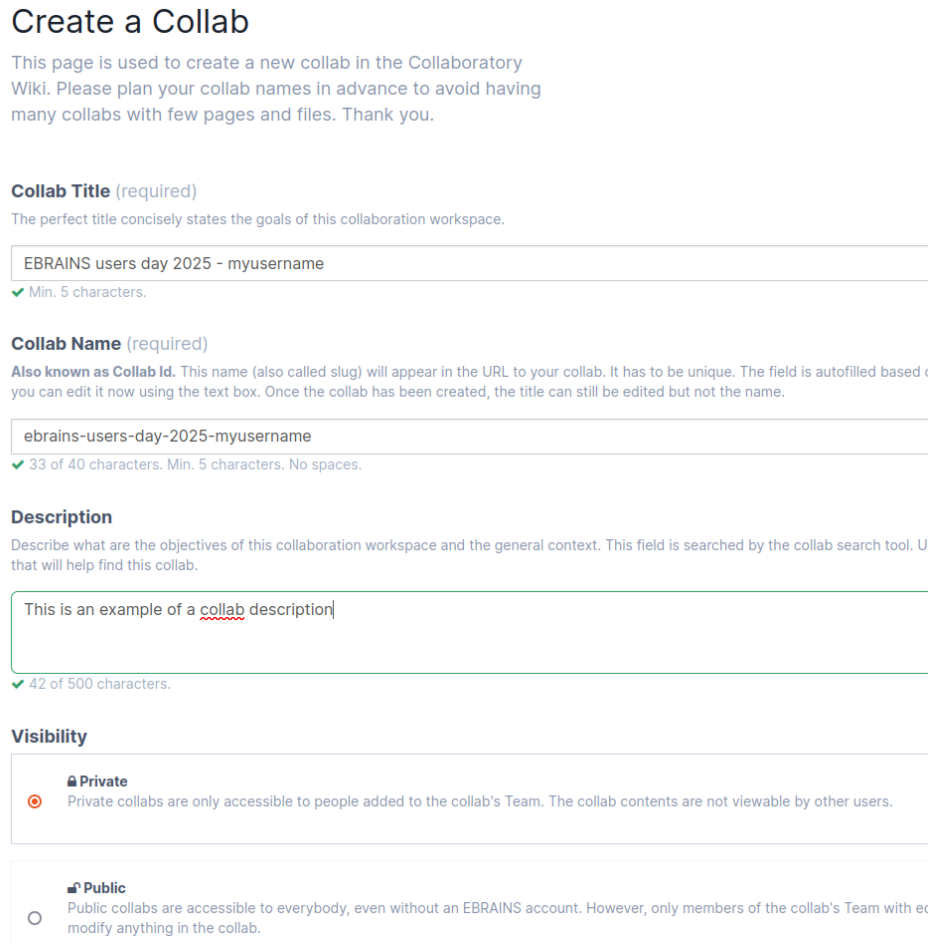
Setup: creating a collab in the Collaboratory

- https://wiki.ebrains.eu



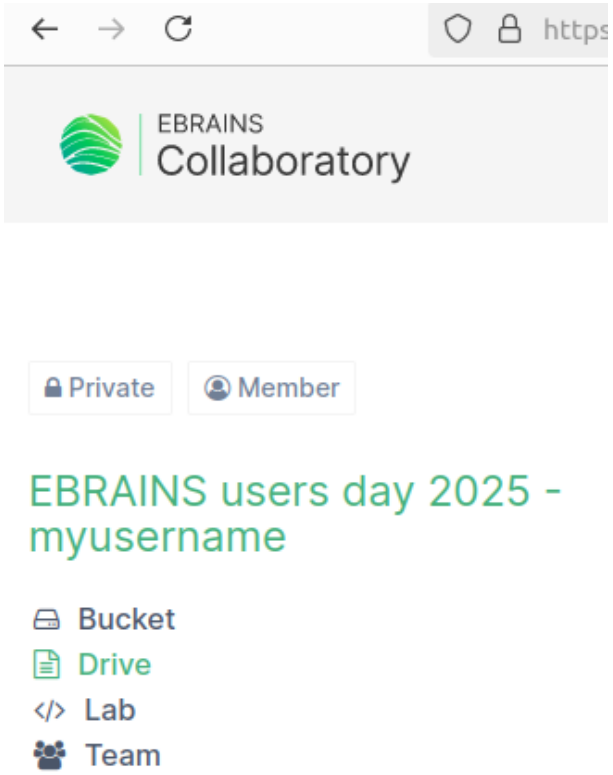
Setup: creating a collab in the Collaboratory

- https://wiki.ebrains.eu



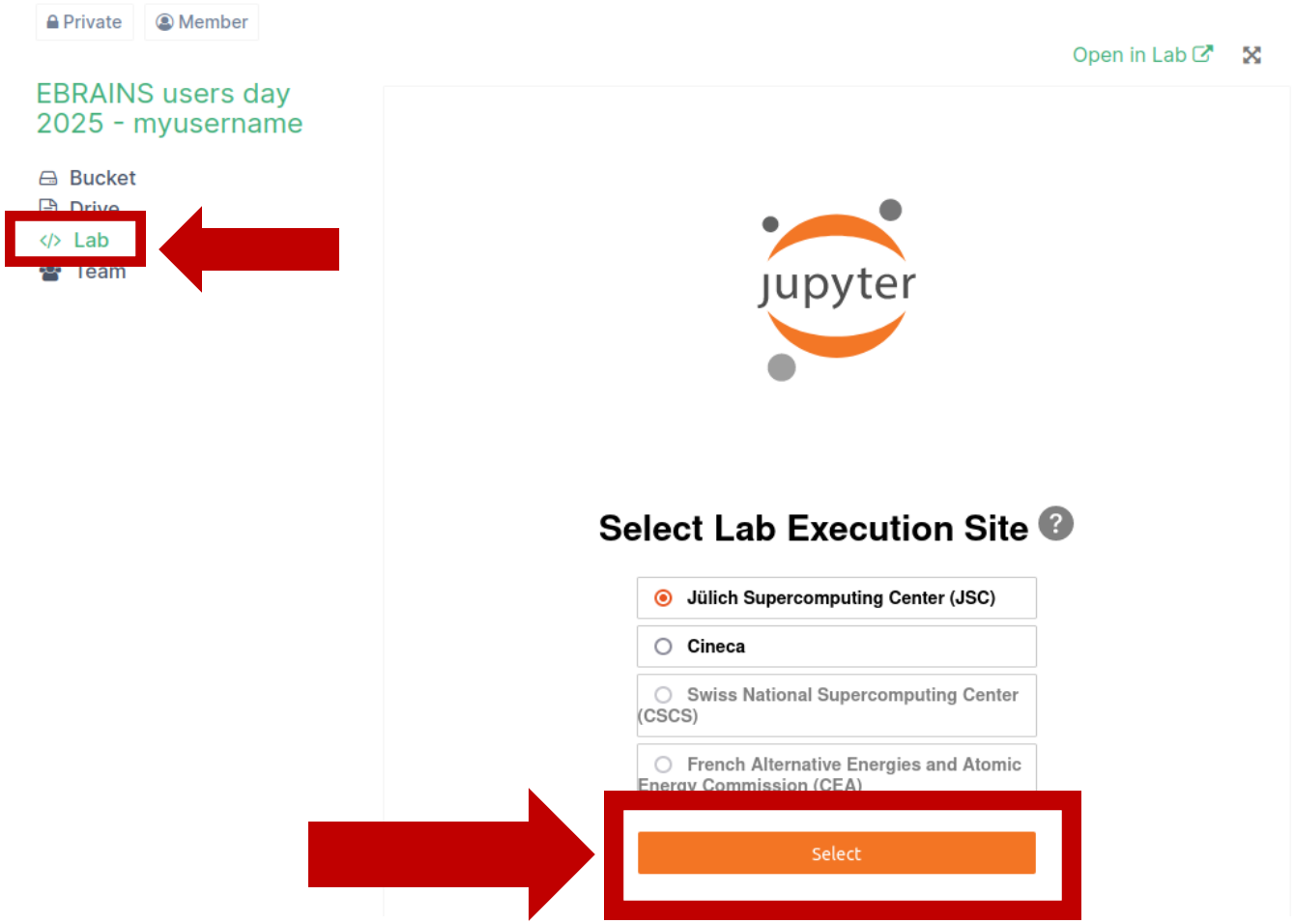
Setup: creating a collab in the Collaboratory

- https://wiki.ebrains.eu

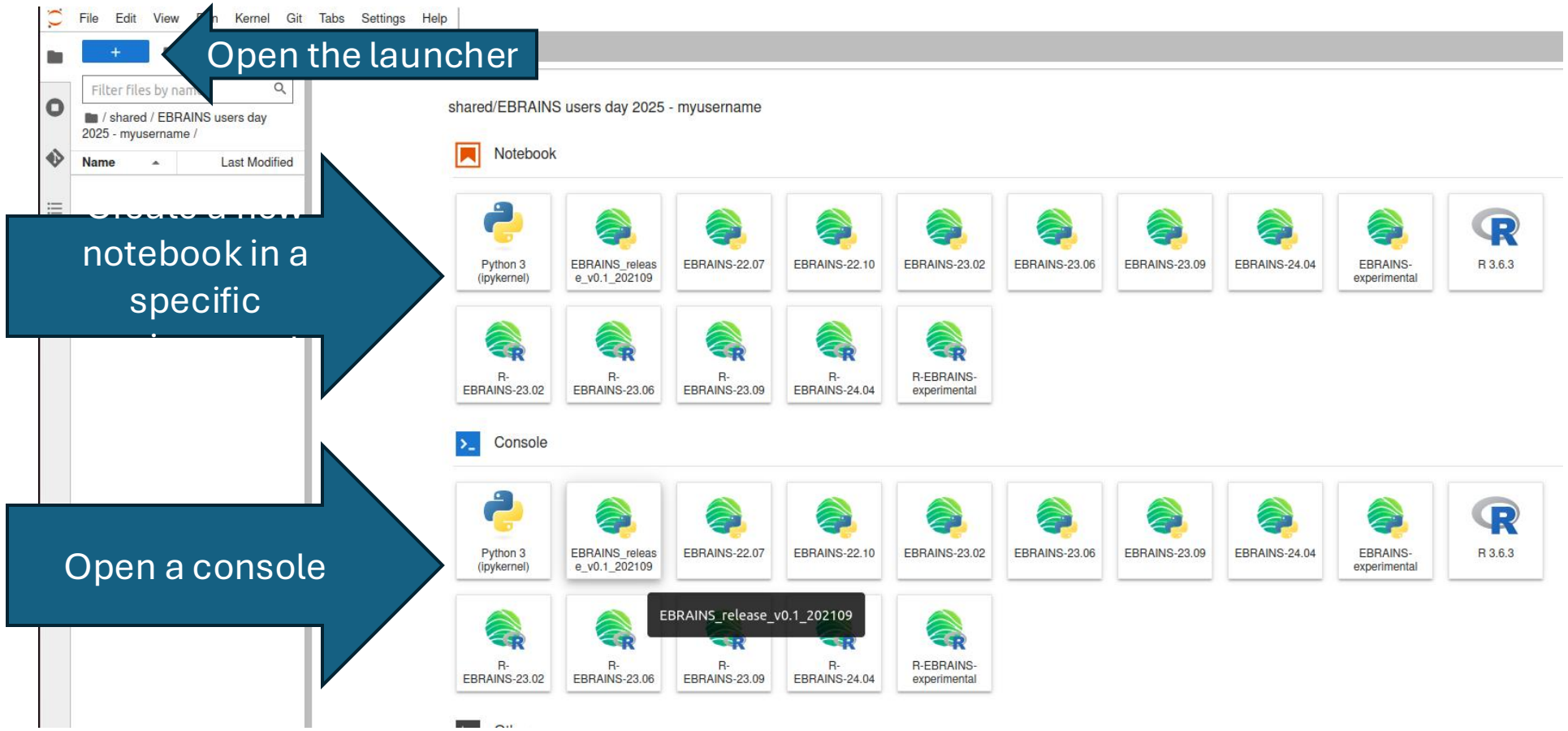


- **Bucket**: large storage space but cannot be edited, ideal for data
- **Drive**: files and documents that may need to be edited or moved
- **Lab**: cloud-based development environment for developing and running programs without needing to install anything
- **Team:** manage team members and their permissions
- Check out the documentation for more information

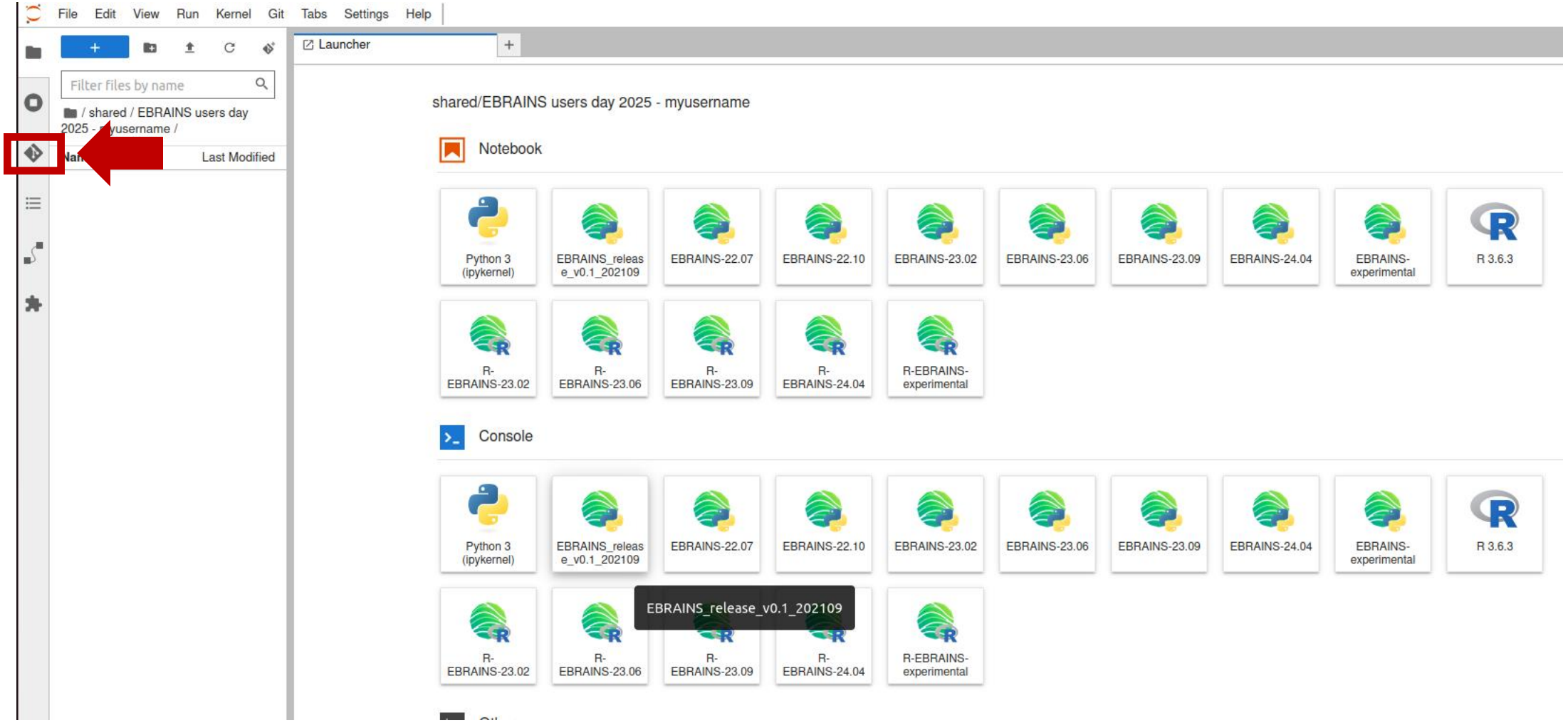
Use the Lab in a collab



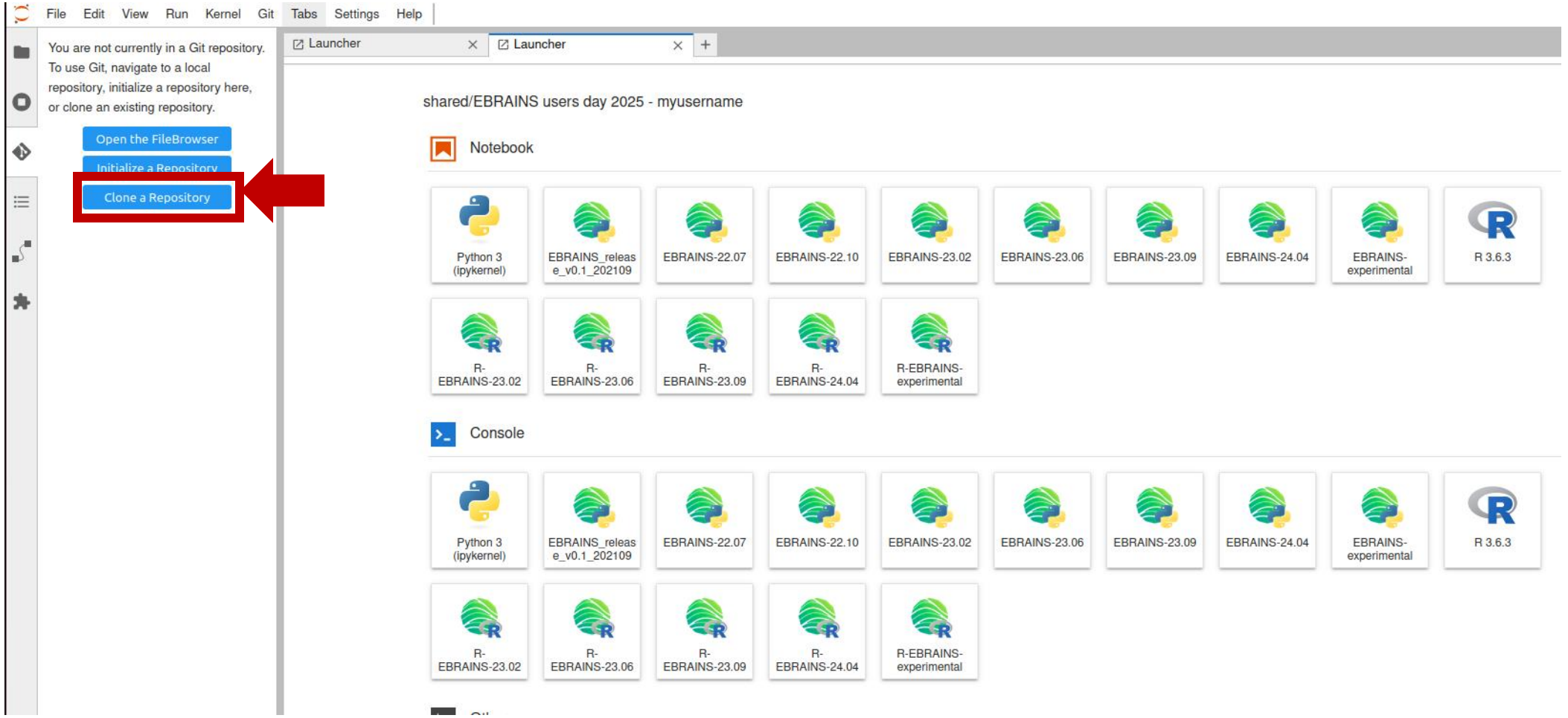
Use the Lab in a collab



Retrieve the tutorial notebooks via Github

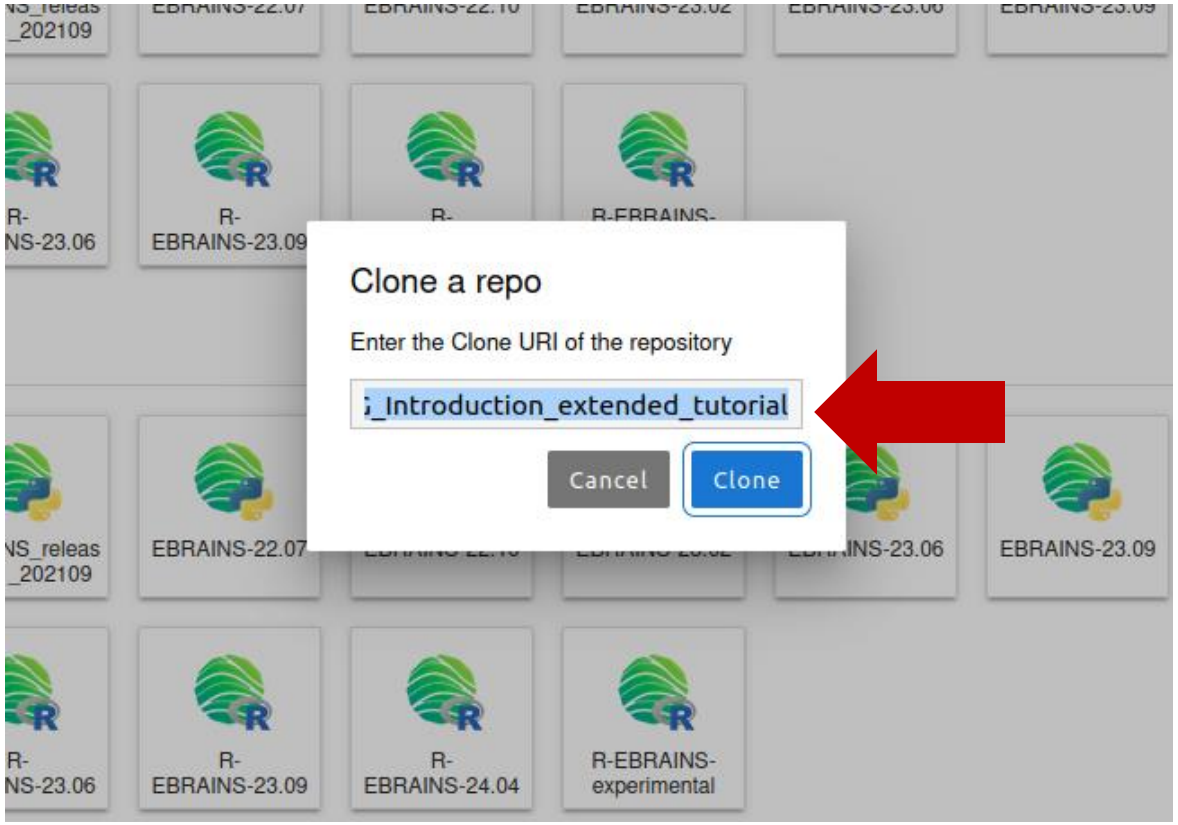


Retrieve the tutorial notebooks via Github



Retrieve the tutorial notebooks via Github

- https://github.com/Alixbonard/EBRAINS_users_day_2025_KG_Introduction_extended_tutorial



2. Searching a dataset with EBRAINS Knowledge Graph Search (KG search)

2. Search a dataset with the Knowledge Graph

Try to search a dataset related to hippocampal CA1 pyramidal neurons.
Requirements:

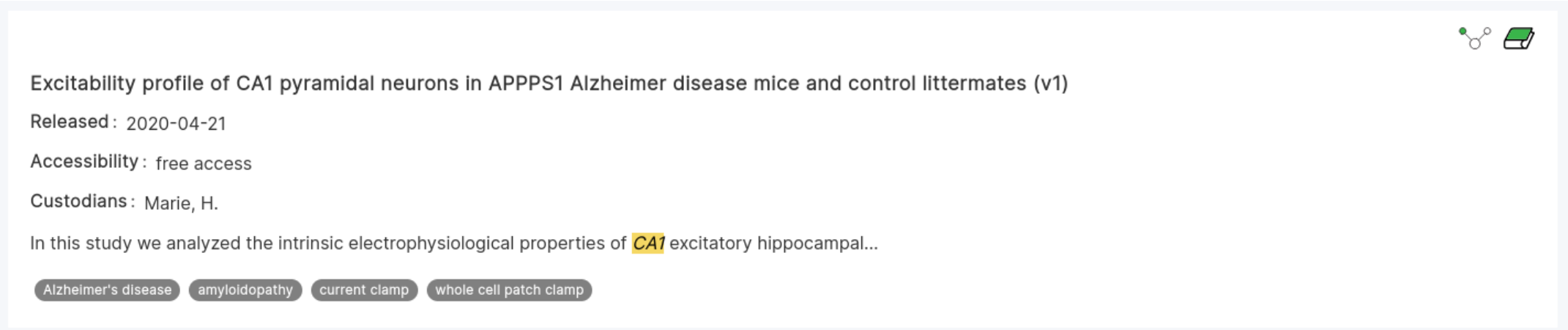
- free access
- 6 months old C57BL/6J mice
- Alzheimer's disease
- Electrophysiology experimental approach

Resources:

https://search.kg.ebrains.eu/

2. Search a dataset with the Knowledge Graph

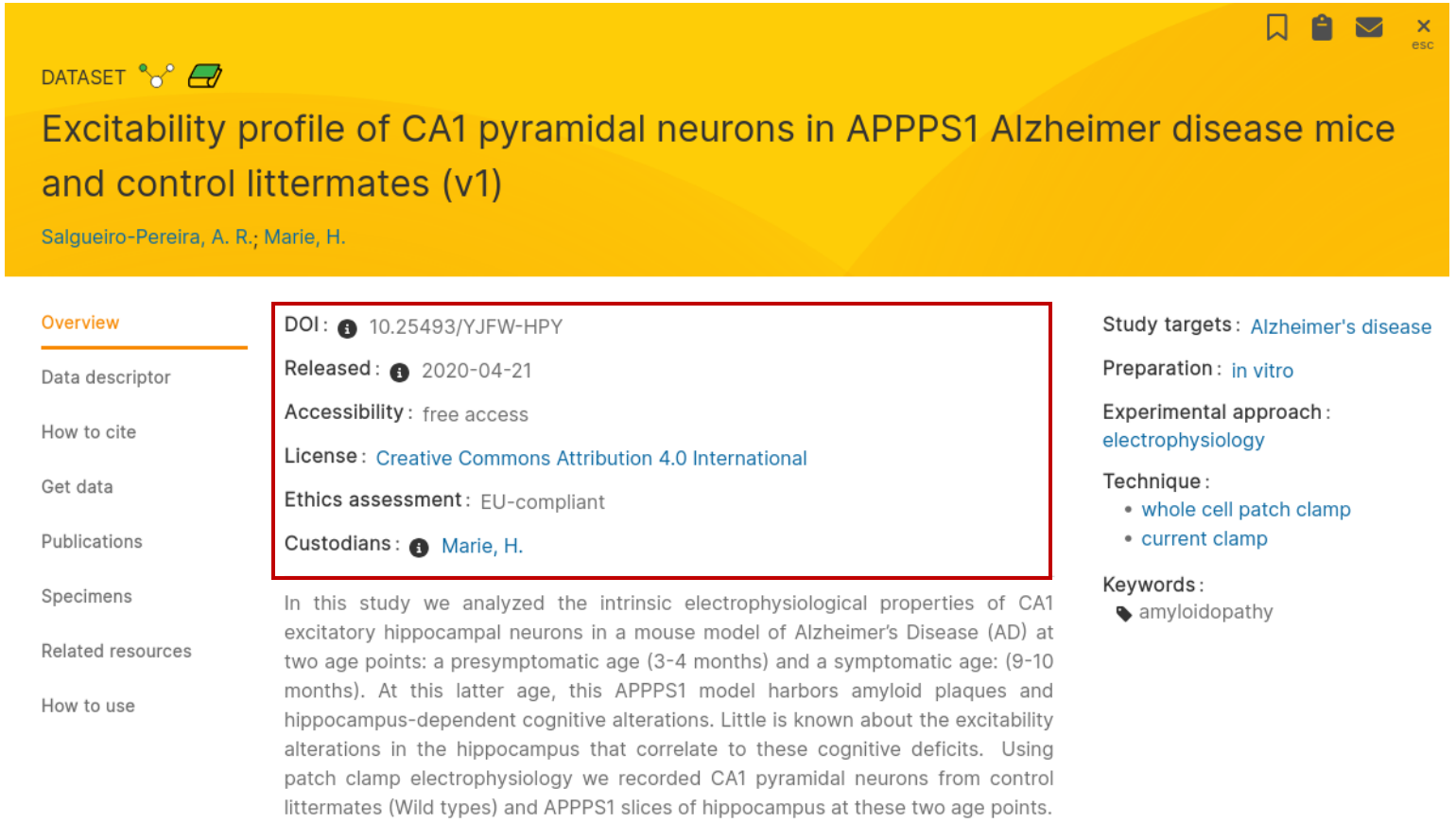
Results:



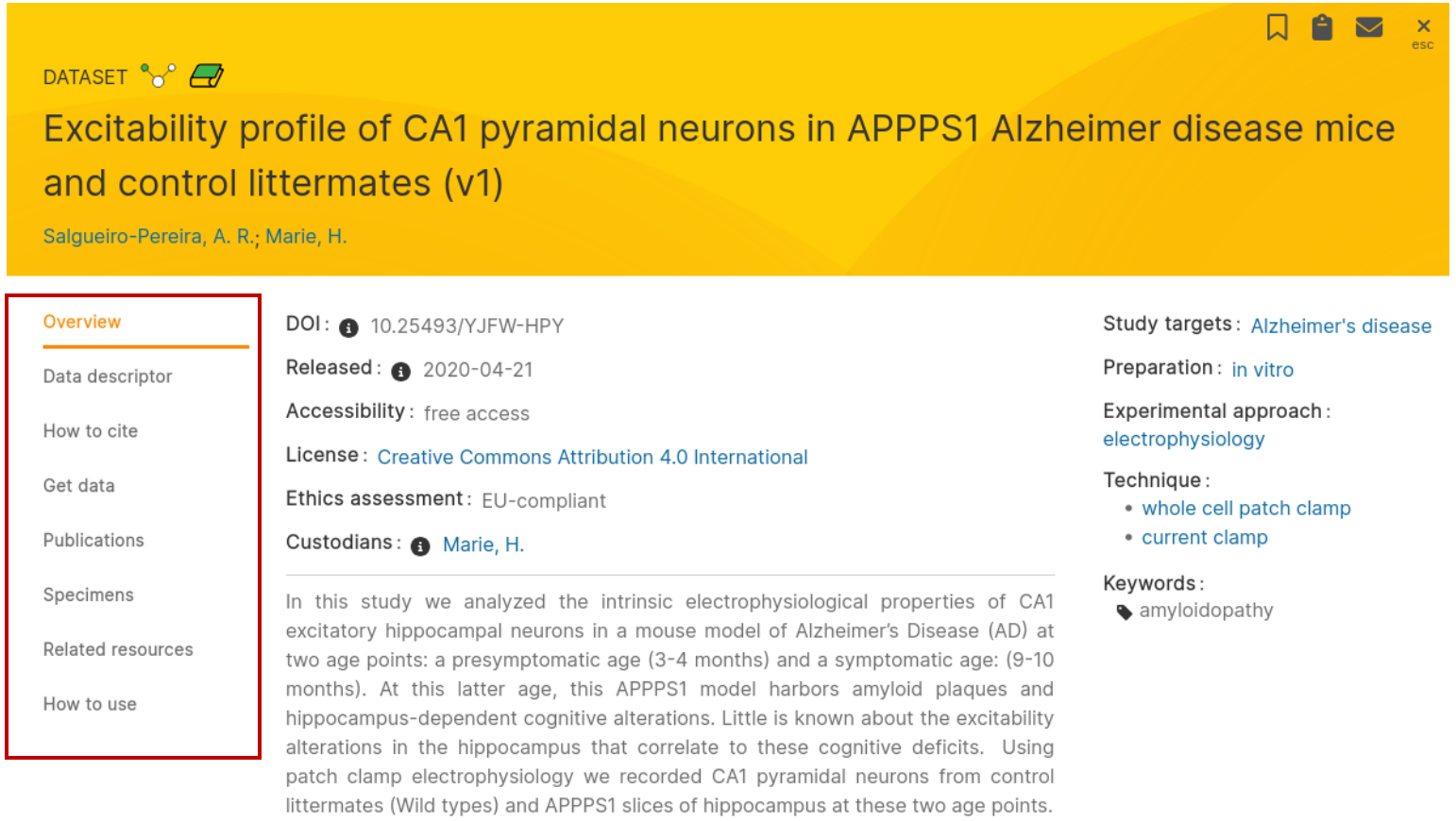
Resources:

https://search.kg.ebrains.eu/

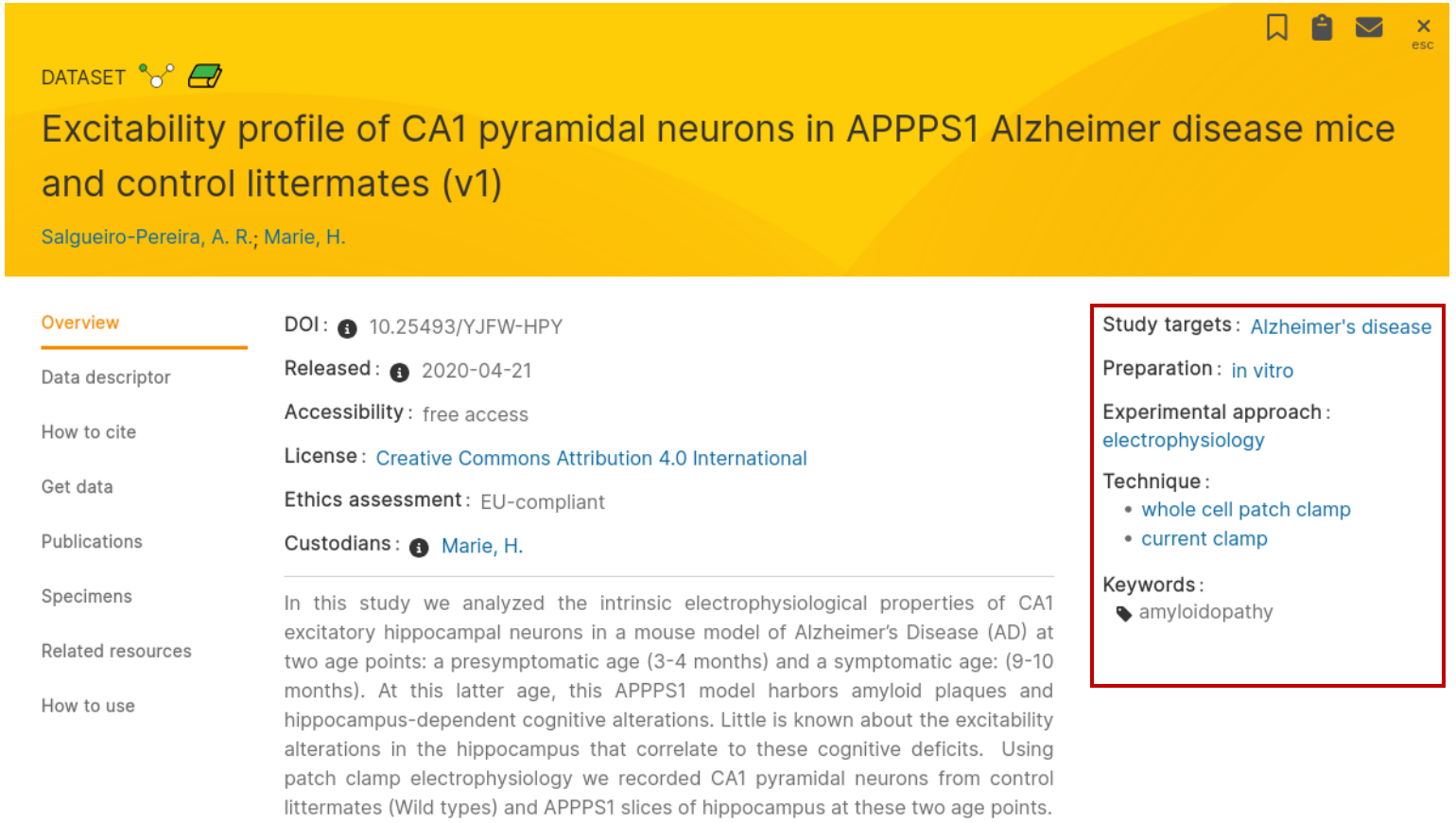
Search a dataset with the Knowledge Graph



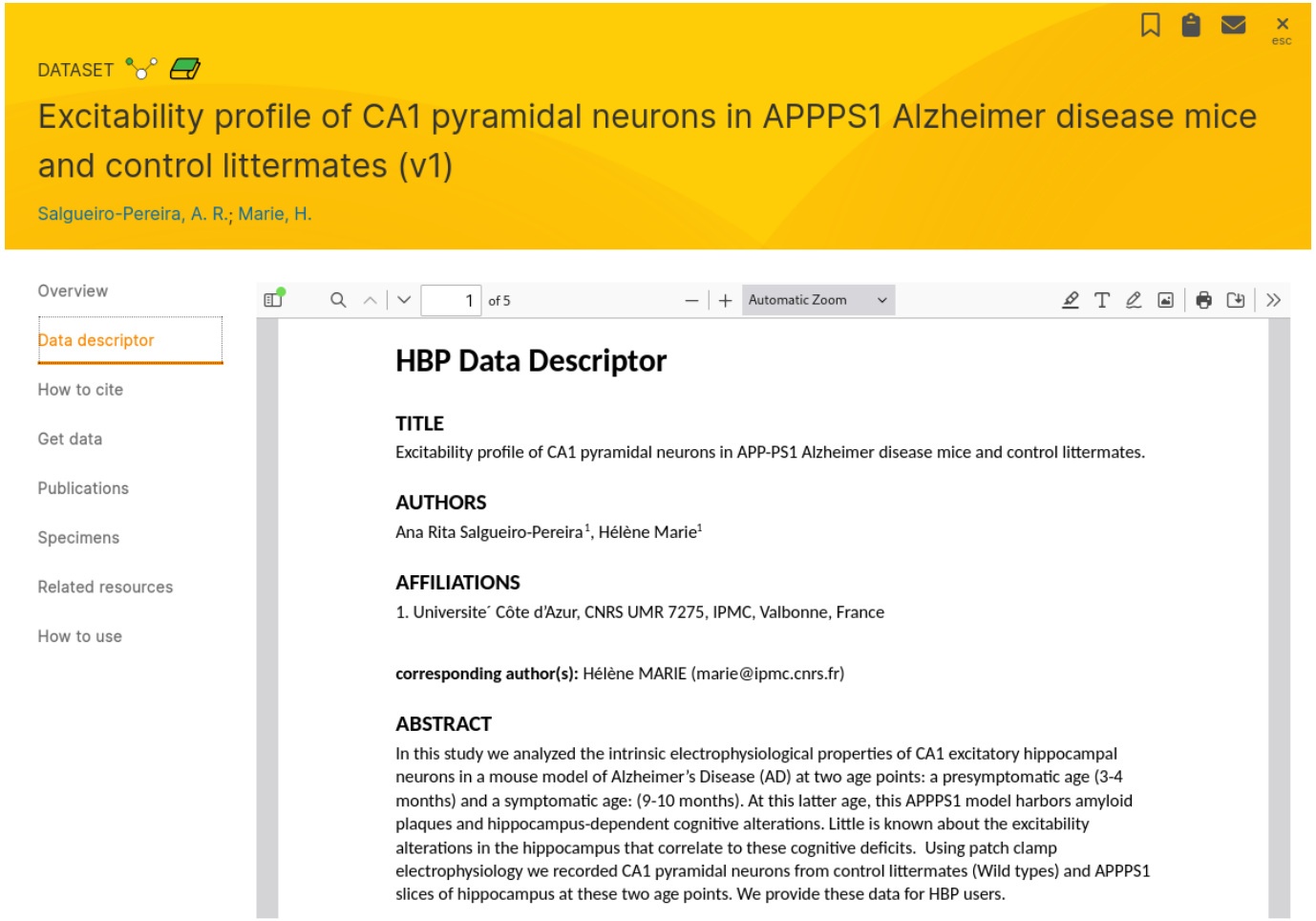
Search a dataset with the Knowledge Graph



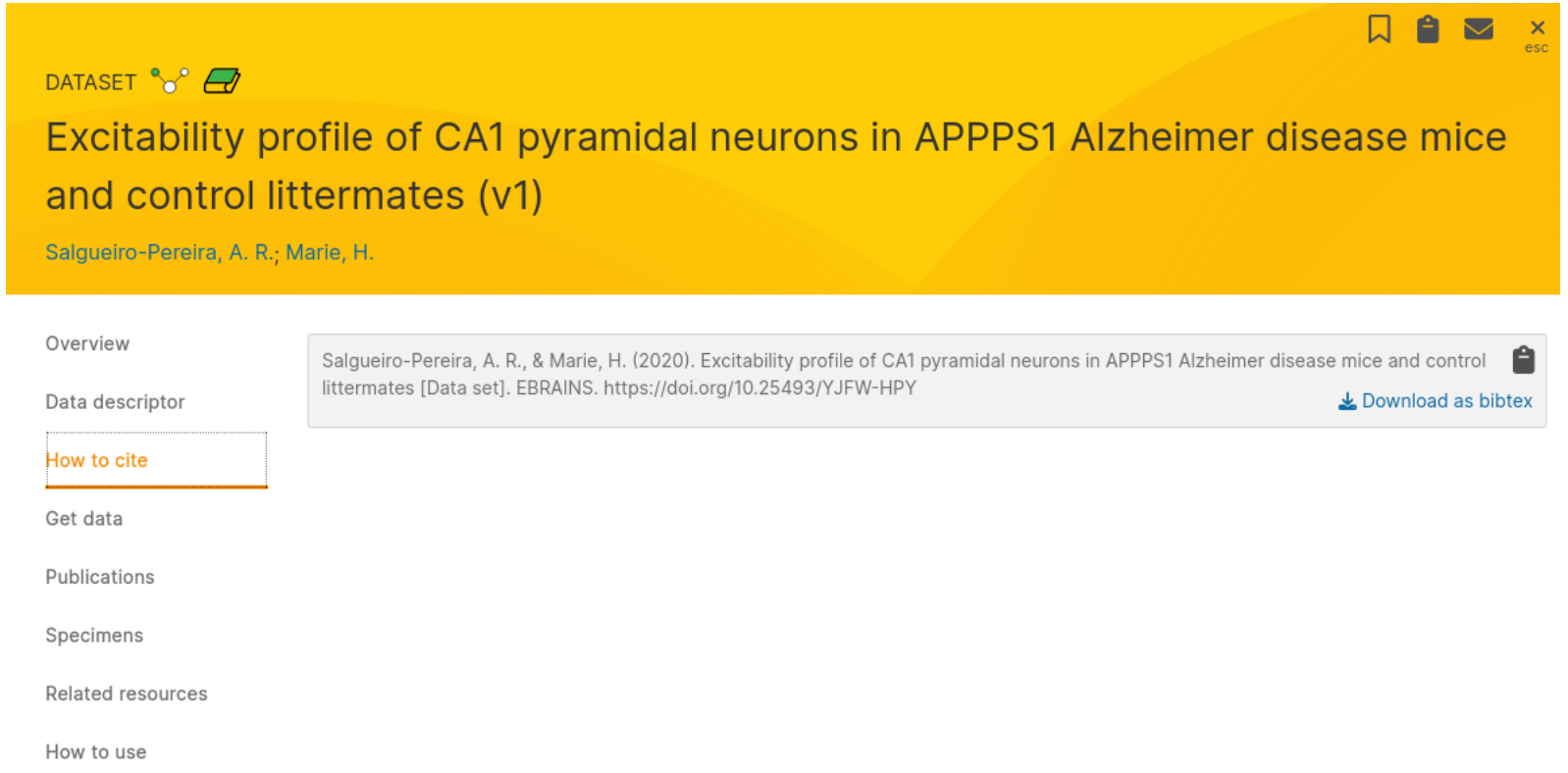
Search a dataset with the Knowledge Graph



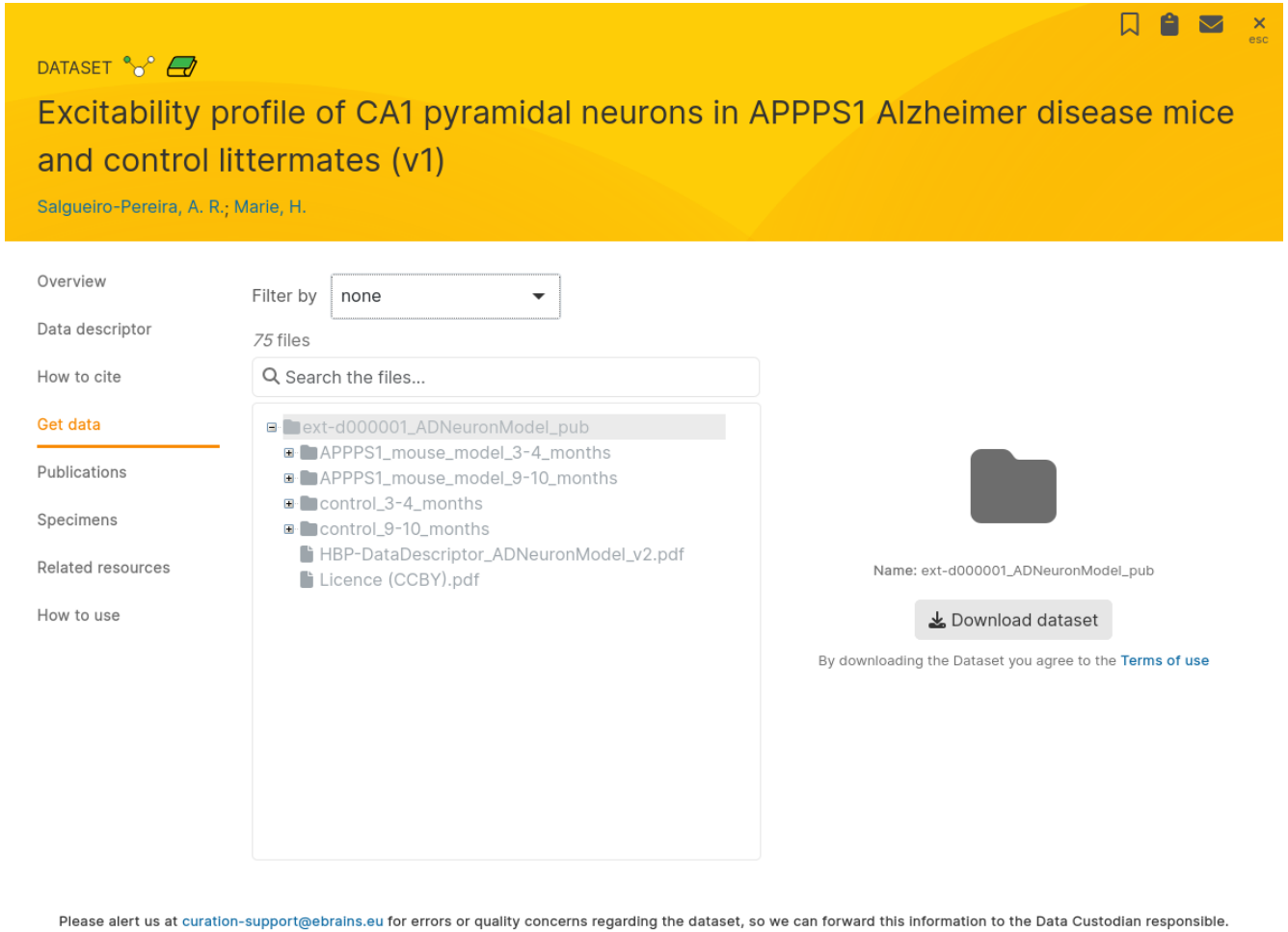
Search a dataset with the Knowledge Graph



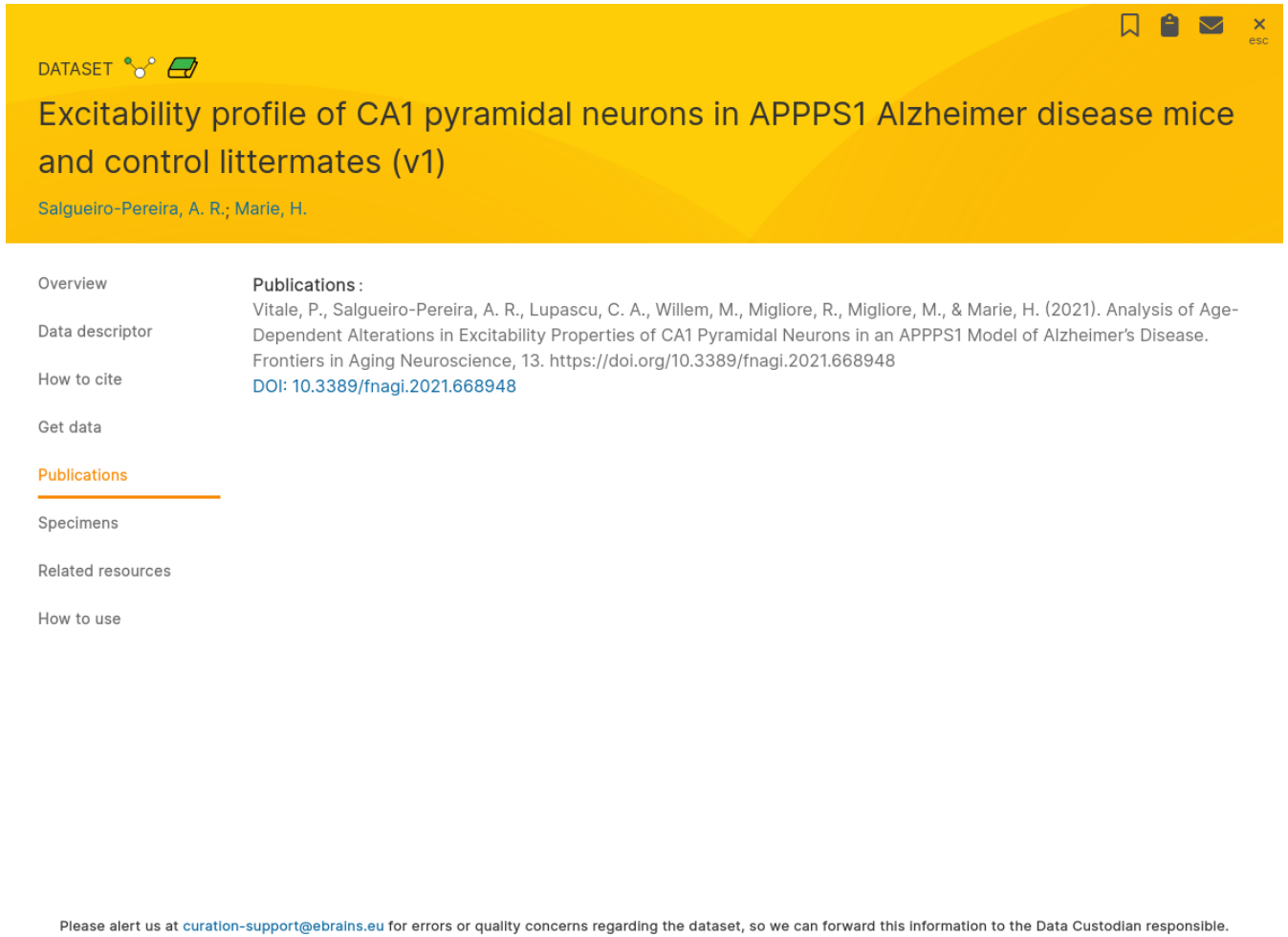
Search a dataset with the Knowledge Graph



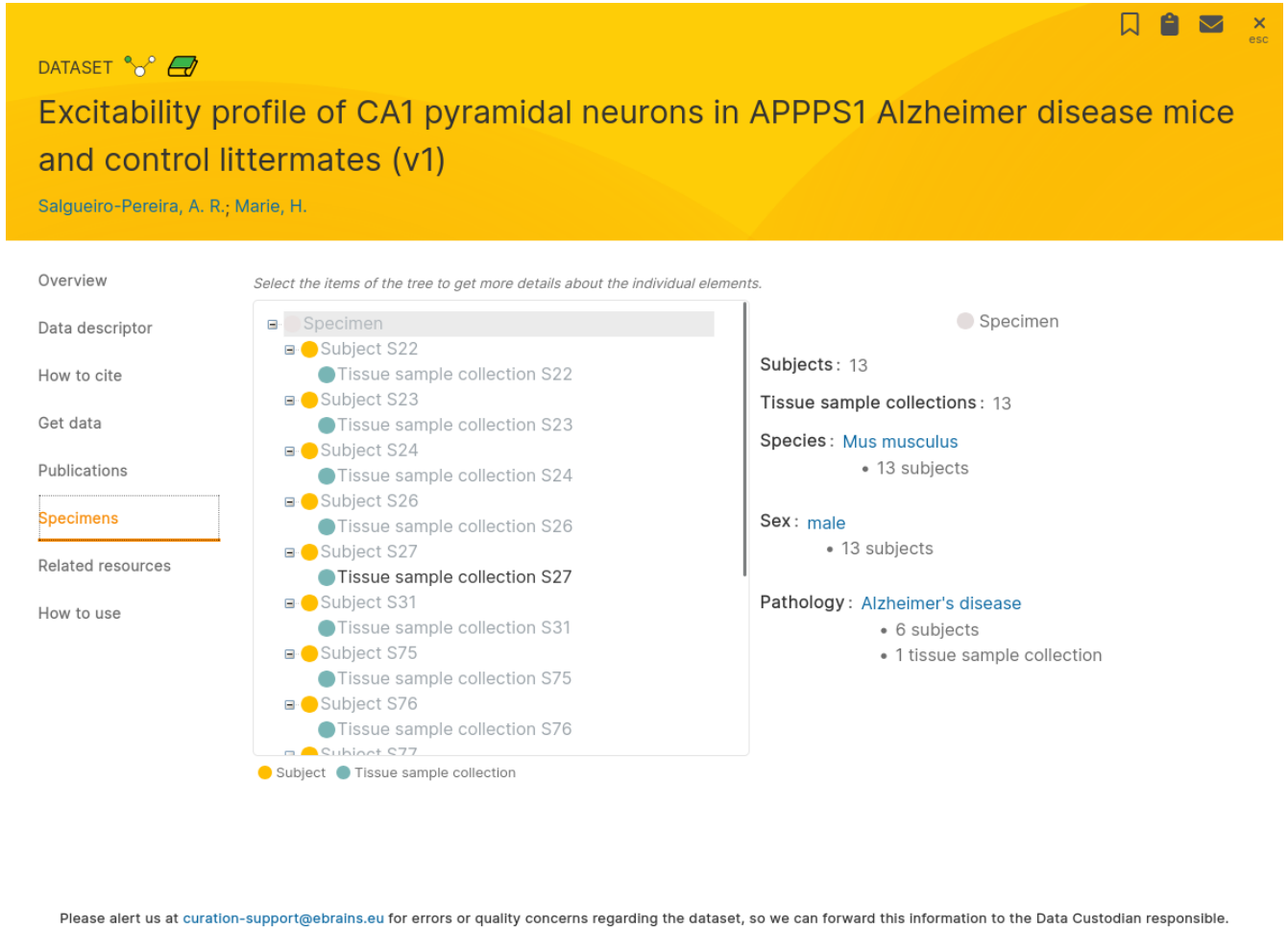
Search a dataset with the Knowledge Graph



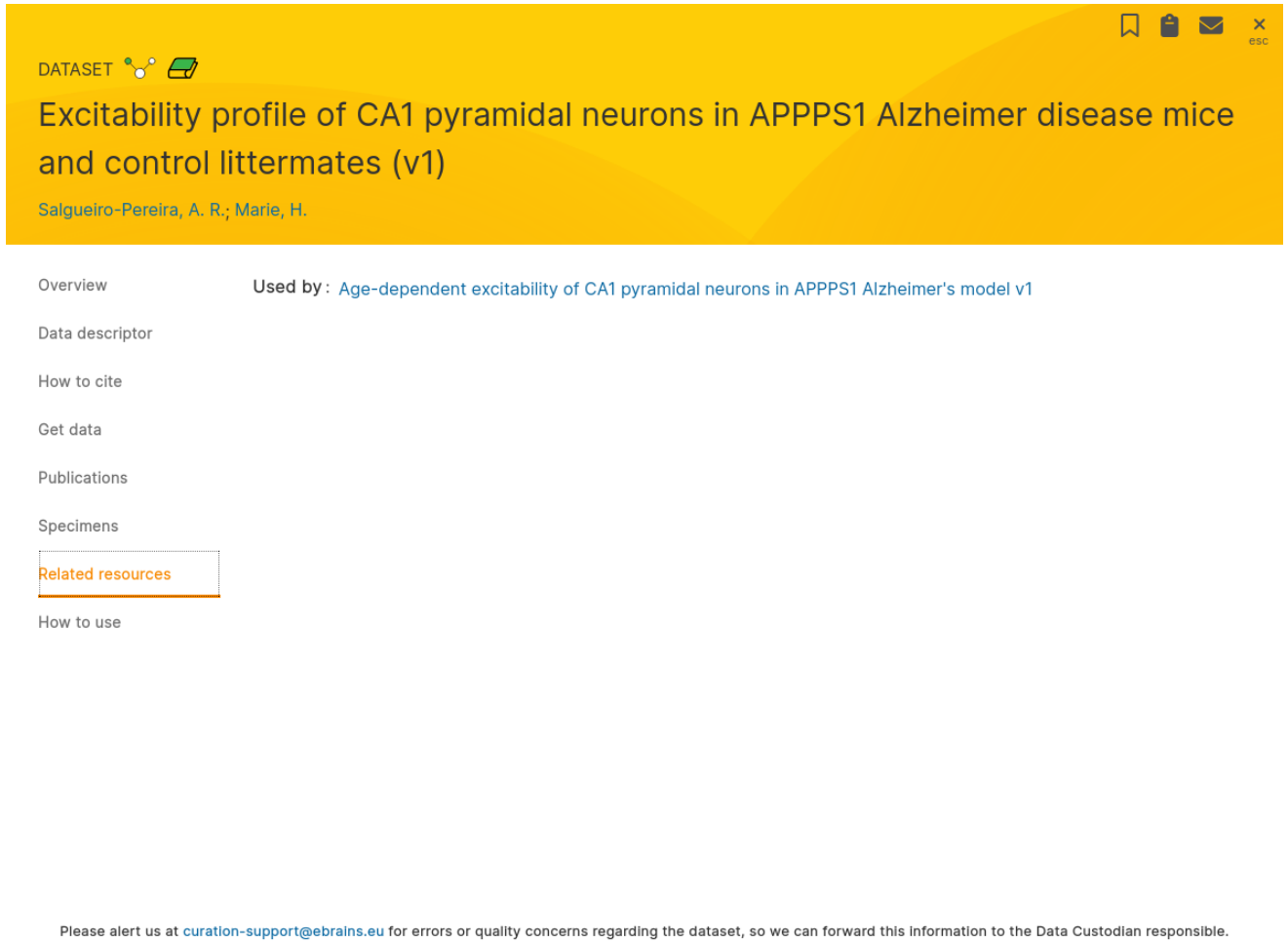
Search a dataset with the Knowledge Graph



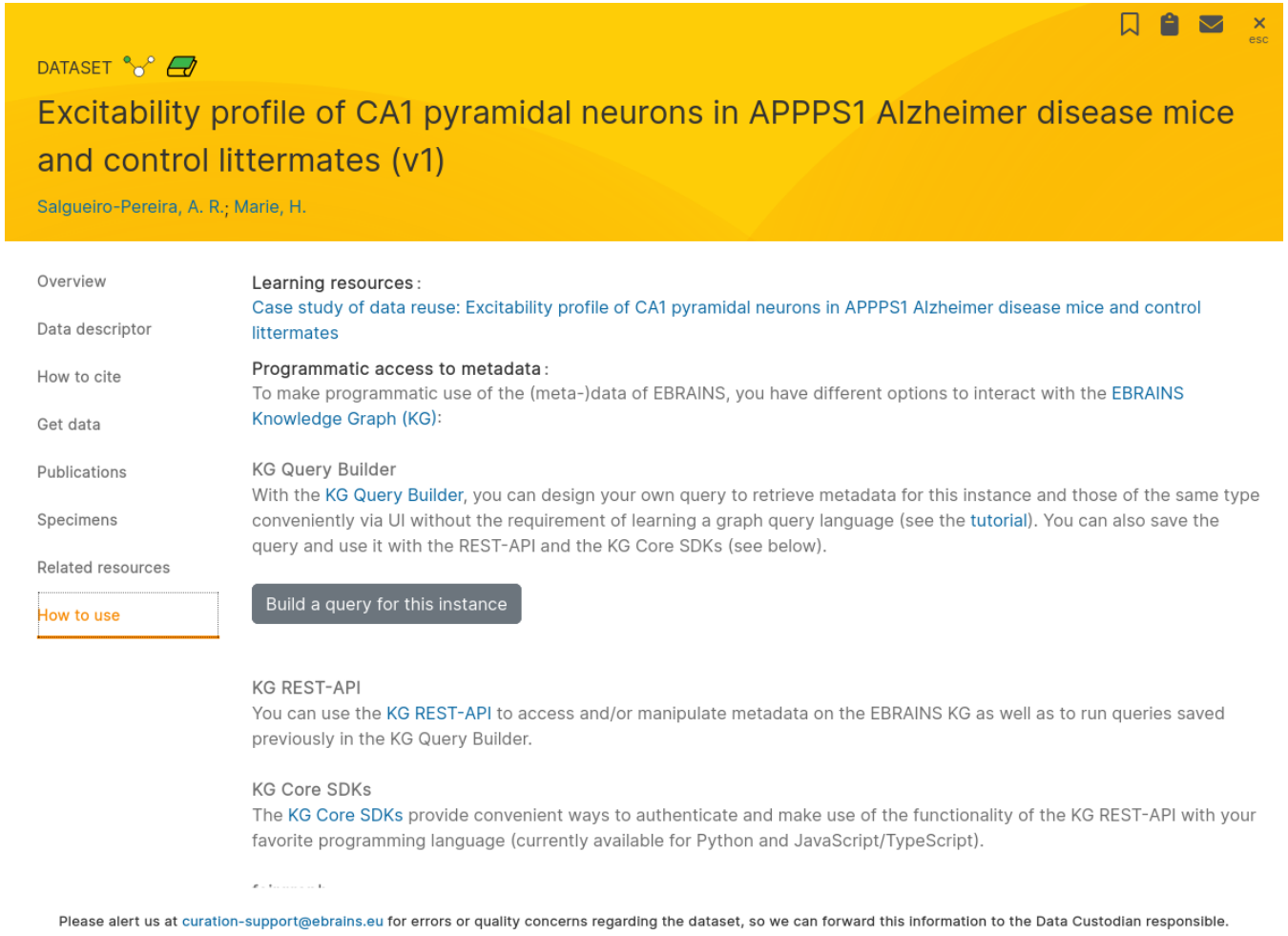
Search a dataset with the Knowledge Graph



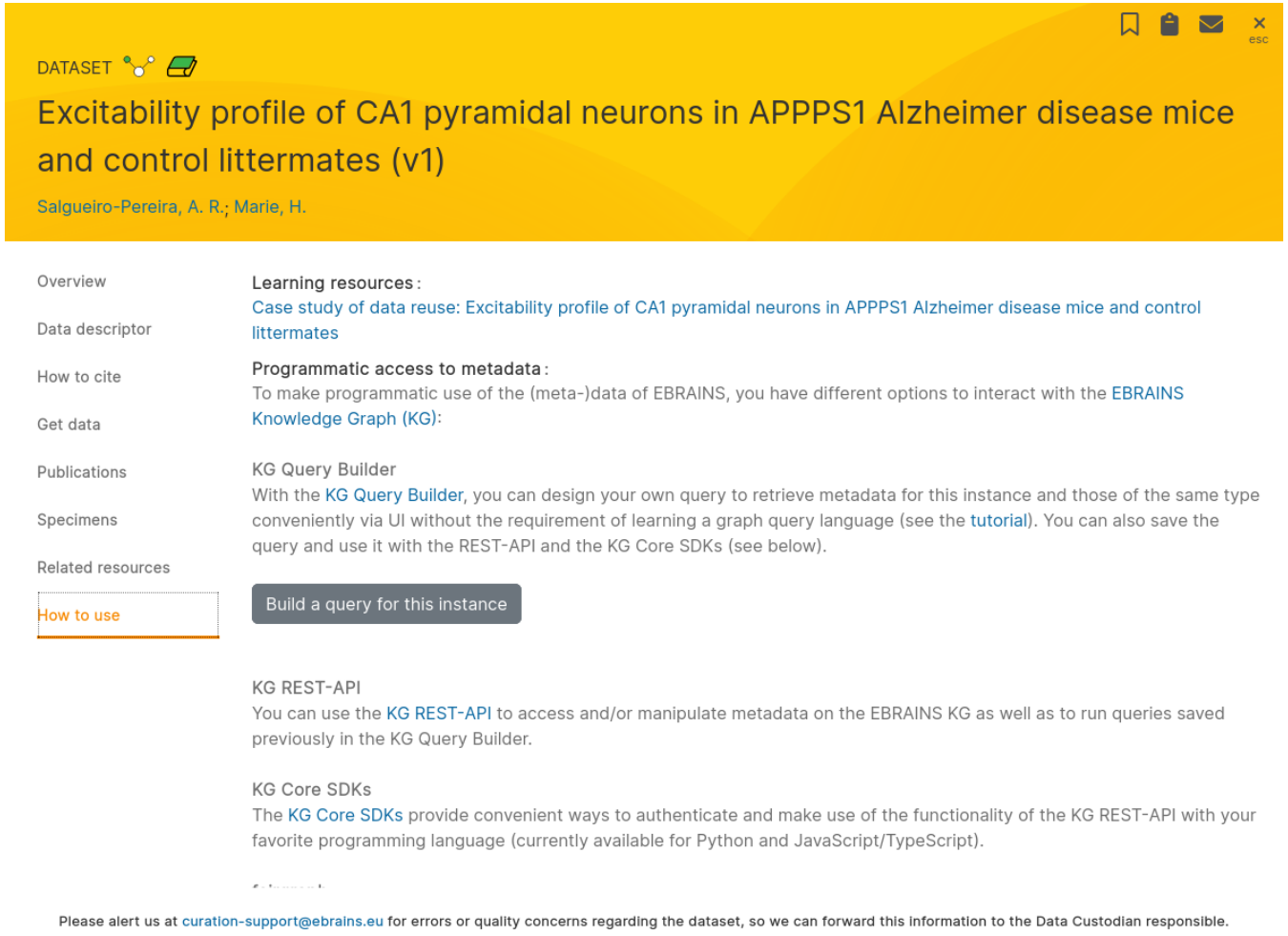
Search a dataset with the Knowledge Graph



Search a dataset with the Knowledge Graph



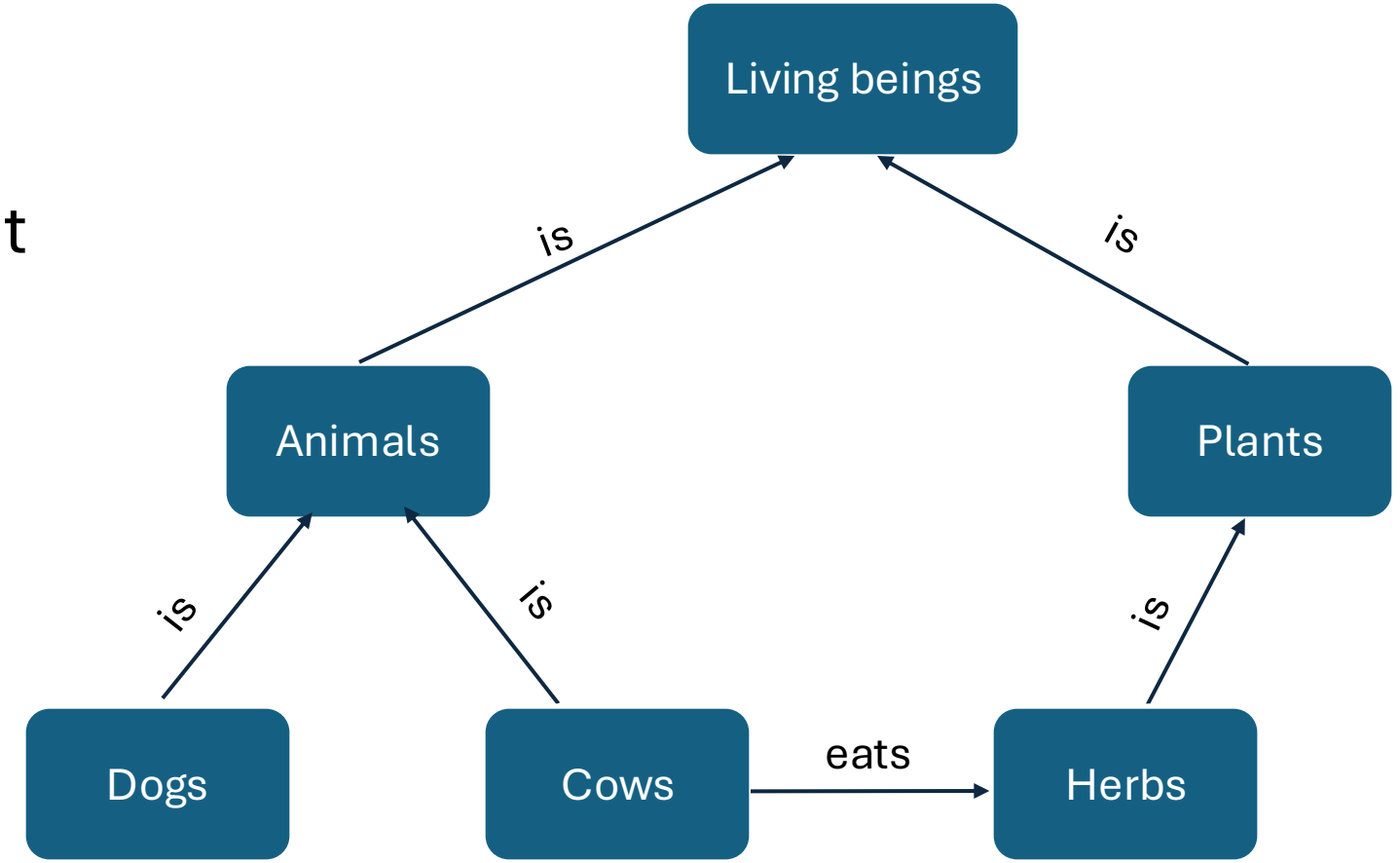
Search a dataset with the Knowledge Graph



3. Retrieving a dataset in a Jupyter notebook with the Fairgraph library

What is a Knowledge Graph?

- A way to represent entities and their relationships



Retrieve datasets and their metadata with Python : the fairgraph library

- Client and authentication token

1. **Client with: JupyterLab**

]:
```
from fairgraph import KGClient

kg_client = KGClient(host="core.kg.ebrains.eu") # if you want to have access to
```

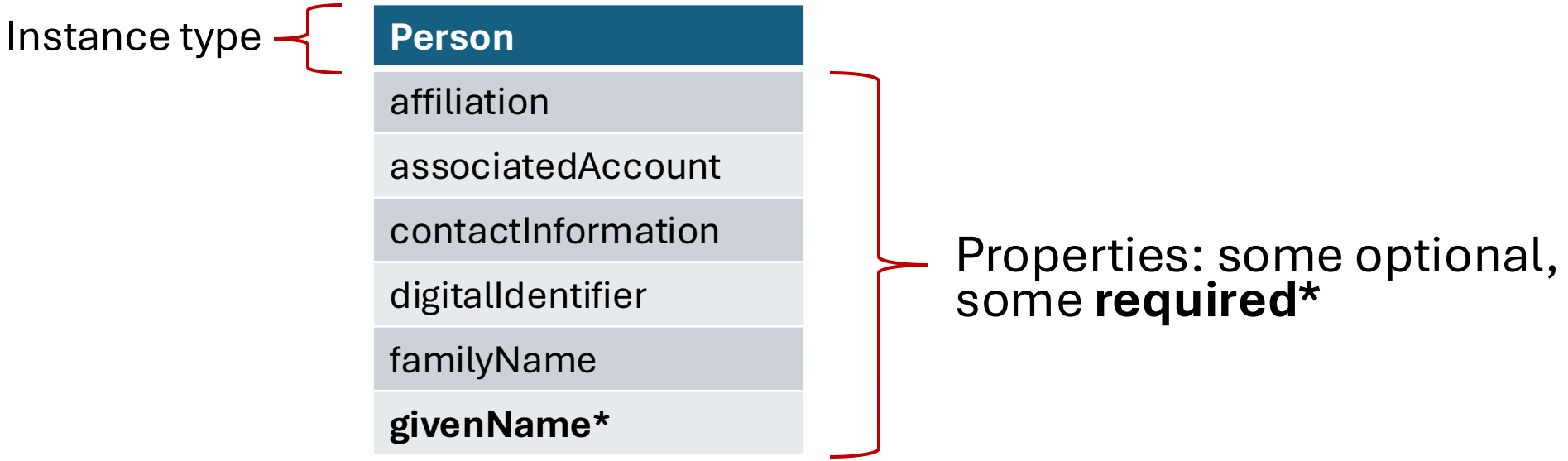
2. **Client with: Local system**

You can have access to your token in the KG Editor in your profile button (copy token to clipboard)

]:
```
# in a local terminal:

# cd </path/to/tutorial/folder> # change the directory where the tutorial file .
# export KG_AUTH_TOKEN=<paste the token> (e.g. 'eyJhbGci...nPq')
#

! pip install fairgraph # to install fairgraph in your python environment

# in your jupyter notebook:

from fairgraph import KGClient
import os

#kg_token = os.environ['KG_AUTH_TOKEN']
#kg_client = KGClient(host="core.kg.ebrains.eu",token=kg_token)

# alternative approach 1:
# run in an another cell - get the token everytime you need
# kg_token = KGClient(host='core.kg.ebrains.eu').user_info()

# alternative approach 2:
# copy your token from the KG Editor directly in the jupyter notebook as below
kg_token = "yourtoken"
kg_client = KGClient(host="core.kg.ebrains.eu",token=kg_token)
```

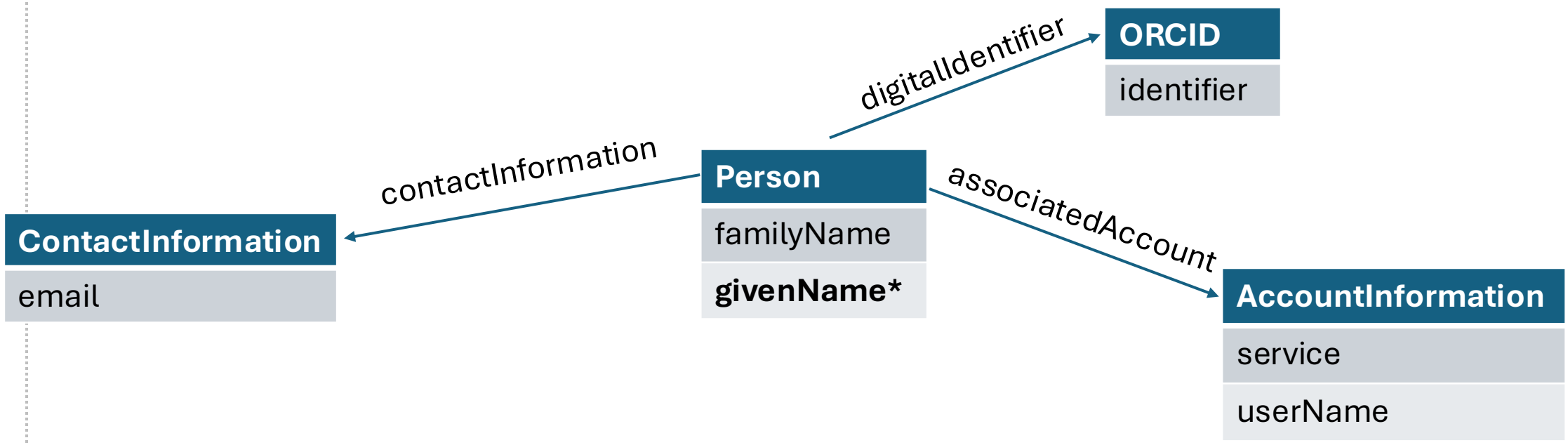
OpenMINDS

- Need to be able to infer information (e.g. a Person must have a name)
- Metadata framework for linked data: openMINDS
- openMINDS contains schema specifications for multiple metadata models



OpenMINDS

- Properties may expect links to other instances



Find schema specifications and more detailed explanations here:
https://openminds-documentation.readthedocs.io/en/latest

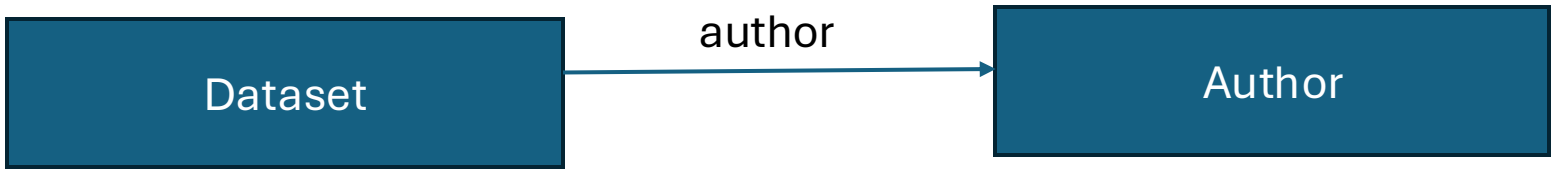
The fairgraph Python library

- Available here: https://github.com/HumanBrainProject/fairgraph/
- Uses a client to access the KG: needs your credentials
  - Retrieve a token from the KG editor or use the EBRAINS Jupyter lab (with a collab, for example)
- Important classes:
  - KGClient
  - KGObject: parent class for every object on the KG
  - KGProxy: representation of a KGObject: type and identifier are known, but not its metadata
  - KGQuery: Representation of one or more KGObjects identified by a range of possible types and by some of their metadata, but whose specific identifier(s) is/are not known.

The fairgraph Python library

- Requires some understanding of openMINDS
- Does not follow links by default

```
Dataset.by_name('Dataset name', kg_client)
```



The fairgraph Python library

- Use the follow_links parameter to follow links

```
Dataset.by_name('Dataset name', kg_client, follow_links = {"authors": {}})
```

Retrieve datasets and their metadata with Python : the fairgraph library

- Retrieving the dataset:
  - Using the dataset name
  - Using the ID
  - Using the DOI

Retrieve datasets and their metadata with Python : the fairgraph library

- Downloading the dataset

Downloading the dataset

Once the dataset retrieved in a fairgraph object, you can download the all dataset into your local system (your working directory for example) you can use the property `download`. To do so, you have to define your `local_path`. Then, you must use download property with the local_path, the kg_client as arguments. You must accept the terms of use as shown below.

```
]:
# Get your local path
import os

local_path = os.path.abspath('path/to/download/directory')
# or
local_path = os.getcwd() # get the working directory
```

```
]:
dataset_version.download(local_path='...', client=kg_client, accept_terms_of_use=True)
```

Exploring dataset metadata

- Authors

Authors

In the previous part, you learned how to retrieve a dataset with `Dataset` or `DatasetVersion` presenting explicit and non-explicit metadata. Here you will learn how to explicit them. To explicitly retrieve authors you have to use the argument `follow_links`. When you fetch a dataset with Dataset or DatatsetVersion, you fetch default nodes (i.e. explicit metadata), with follow_link you add more nodes. Thereby, you can access the target metadata. Let's try with authors:

You want to retrieve the names of the dataset's authors. By consulting the Dataset schema specifications in the openMINDS documentation, you can see which link you need to follow to obtain that piece of information.

[ ]:

```
# Retrieve the dataset object:
dataset = Dataset.by_name('Excitability profile of CA1 pyramidal neurons in APPPS1 Alzheimer disease mice and cor
print(f'This dataset has {len(dataset.authors)} authors. {dataset.authors[0].family_name} {dataset.authors[0].giv
```

Exploring dataset metadata

- **Exercise**: Find the following metadata:
  - Datafiles
  - Publication
  - DOI
  - Experimental Approach
  - Techniques
  - Studies Specimens

Finding datasets with fairgraph queries: A brief introduction

4. Finding datasets with fairgraph queries: a brief introduction

This part is a brief introduction to queries. Please see the tutorial EBRAINS Knowledge Graph Advanced for more information. One other use of Fairgraph library is the Queries. Queries are useful if you need to search a dataset using filters. As presented above, you can use `DatasetVerison` to obtain a list of potential datasets of interest. For more information you can consult this link

Let's see an example.

Run this cell to obtain datasets having "neurons" in their name:

In [ ]:
```
datasets = DatasetVersion.list(kg_client, name="neurons")
for dataset in datasets:
    print(dataset.name)
```

Let's try "patch-clamp" technique:

In [ ]:
```
datasets = DatasetVersion.list(kg_client, name="patch-clamp")
for dataset in datasets:
    print(dataset.name)
```

As you see, you can obtain datasets corresponding to your filter. Fairgrap library is still in development, so queries may not run properly. You can use the EBRAINS query builder to directly build queries and run them on the KG. For more information regarding this tool, click here.

A space just for your notes:

Exercise in autonomy

5. Exercise in autonomy

For this last part, you will have to reproduce what you have learned in the previous parts. Remember that you can look up schema specifications on the openMINDS documentation.

List of dataset:

- 10.25493/YJFW-HPY
- 10.25493/VAV5-BXU
- 10.25493/CHJG-7QC
- 10.25493/M1AQ-3AC
- 10.25493/M1V0-WE3
- 10.25493/3NTS-Q0B
- 10.25493/JNFA-HDP
- 10.25493/5GE0-6MF

1. Exercise: Use fairgraph query to find a dataset of interest

2. Exercise: Explore the dataset metadata. Can you retrieve the authors? See if there is more datasetversion? Retrieve the date of publication? The publication DOI? You can start with these questions and explore more relevant metadata

3. Exercise: Retrieve dataset files. Can you obtain the name of each file?

4. Exercise: Download the dataset

The EBRAINS Knowledge Graph

Extended Tutorial: Practice case studies for data reuse

Laura Morel, Alix Bonard, Peyman Najafi

NeuroPSI Paris-Saclay Institute of Neuroscience, CNRS, Université Paris-Saclay, France.

12 March 2025 – Accelerate Your Neuroscience Research with EBRAINS: EBRAINS tutorials and users day, Heidelberg (Germany)

Introduction

Table of contents:

Hands-on case study: Figure reproduction with and without metadata

- Explore the datafile
- Reproduce the figure 4C from The microcircuits of striatum in silico. Proceedings of the National Academy of Sciences (DOI: 10.1073/pnas.2000671117) without metadata
- Reproduce the same figure 4C with metadata

Hands-on a learning resource: Salgueiro-Peirera-Marie-2018 (optional)

- Explore the KG Search to find a dataset with fairgraph queries find the Salgueiro Pererra Marie case study
- Follow the case study

Finding a dataset with fairgraph queries

- **Authentication / authorization**

1. **Client with: JupyterLab**

```
]:
from fairgraph import KGClient

kg_client = KGClient(host="core.kg.ebrains.eu") # if you want to have access to
```

2. **Client with: Local system**

You can have access to your token in the KG Editor in your profile button (copy token to clipboard)

```
]:
# in a local terminal:

# cd </path/to/tutorial/folder> # change the directory where the tutorial file
# export KG_AUTH_TOKEN=<paste the token> (e.g. 'eyJhbGci...nPq')
#

! pip install fairgraph # to install fairgraph in your python environment

# in your jupyter notebook:

from fairgraph import KGClient
import os

#kg_token = os.environ['KG_AUTH_TOKEN']
#kg_client = KGClient(host="core.kg.ebrains.eu",token=kg_token)

# alternative approach 1:
# run in an another cell - get the token everytime you need
# kg_token = KGClient(host='core.kg.ebrains.eu').user_info()

# alternative approach 2:
# copy your token from the KG Editor directly in the jupyter notebook as below
kg_token = "yourtoken"
kg_client = KGClient(host="core.kg.ebrains.eu",token=kg_token)
```

Finding a dataset with fairgraph queries

Fairgraph Query

You need to retrieve a dataset related to striatal interneurons with low spiking threshold with `DatasetVersion.list`

In [ ]:
```
"Paste your code here"
```

Download the dataset

To be able to use the dataset, you need to download it. You can use the property `download` of DatasetVersion. Don't forget to use follow_links to retrieve the dataset files.

In [ ]:
```
"Paste your code here"



with zipfile.ZipFile(dataset_path, "r") as z:
    z.extractall("downloads")
```

Checking the dataset - Solution

Run the following cell to check the dataset and to download it.

In [ ]:
```
import os
import sys
sys.path.insert(0, os.path.join(os.getcwd(), ".local"))
```

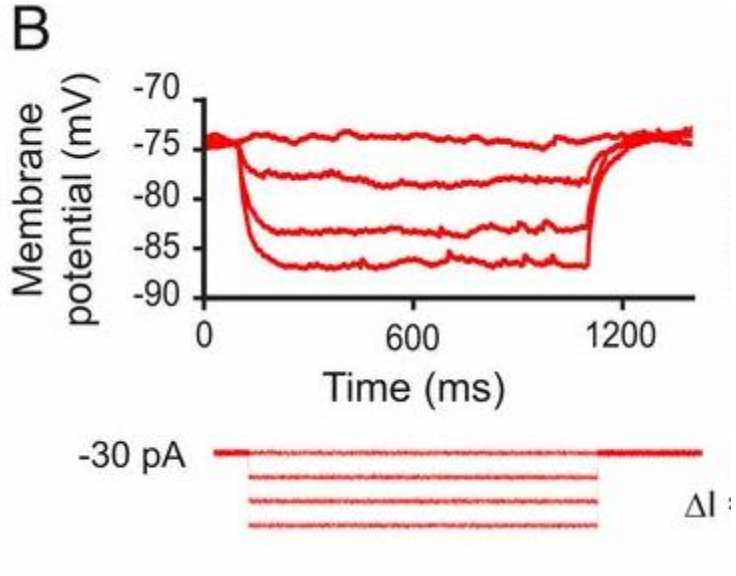
In [ ]:
```
import fairgraph.openminds.core as omcore
import logging
import warnings
import zipfile
```

2. Exploring and visualizing the data

- **Exploring the retrieve dataset**
  - **Exercise**: Explore the data descriptor to understand the dataset structure

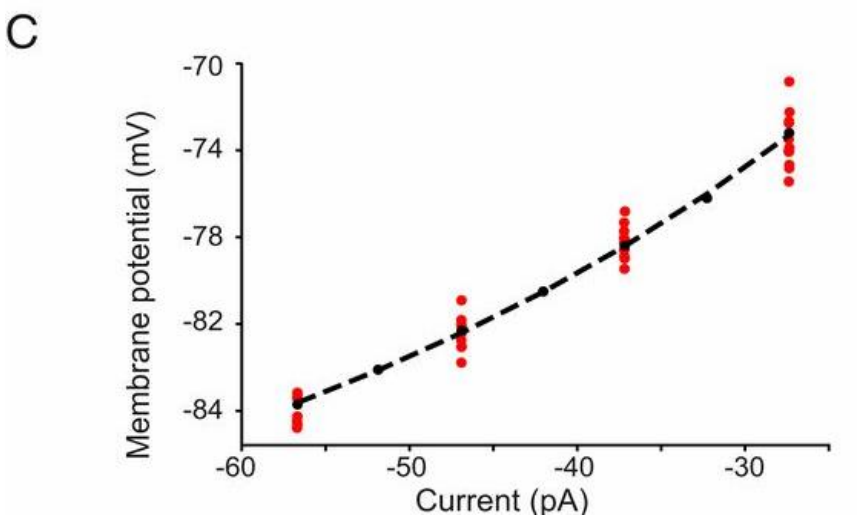
2. Exploring and visualizing the data

- **Exploring the retrieve dataset**
  - **Exercise**: Explore the data descriptor to understand the dataset structure
  - Figure 4B:



3. Reproducing figure without metadata

- **Exploring the retrieve dataset**
  - **Exercice**: Modify the code below in order to reproduce the Figure 4B.



4. Reproducing figure with metadata

Retrieving in-depth metadata

4. Reproducing figure with metadata

With metadata some step presented above can be skip. Let's have a look. The first step was to find current amplitudes values. This step could be very challenging if you don't know how to start and retrieve information as currents. With metadata, you can have a beginning of information that guides you toward figure reproduction. This dataset is quite challenging, therefore, you still need to go through all the membrane potential retrieving steps. However, it is simplifying the process.

Retrieving in-depth metadata

The aim of this part, is to show you why metadata are important for data reuse and how we can use them.
This part might be scary for some of you. In the future, the in-depth curation team will provide more tutorials to retrieve in-depth metadata. This part is an example of in-depth metadata retrieving. If you want to play around, do. If you don't want, just run the code.

The objective is to retrieve the current amplitudes.

```
[ ]:
import fairgraph.openminds.core as omcore
import fairgraph.openminds.stimulation as omstim
import fairgraph.openminds.ephys as ephys
import json


follow_links = {"has_parts" : {}} # create a short cut with dataset_version and stimulation activity

dataset_version = omcore.DatasetVersion.list(kg_client, digital_identifier__identifier = '10.25493/5GE0-6MF', fol

# Part 1: in-depth metadata retrieving

stimulation_activities = [part for part in dataset_version.has_parts if isinstance(part, omstim.StimulationActivi
stimuli = []

for stimulation_activity in stimulation_activities:
# We find every unique EphysStimulus associated with the dataset
    stimulus = stimulation_activity.stimuli.resolve(kg_client)
    if stimulus not in stimuli:
        stimuli.append(stimulus)
# print(stimuli)

specification_json = stimuli[0].specifications.resolve(kg_client).configuration # We retrieve the configuration i
specification = json.loads(specification_json) # We convert the JSON string into a python dictionary
l1_current_value = specification["amplitudes"]
print(f'current amplitudes: {l1_current_value}')
```

Conclusion