May 23, 2024 Version 3

IRIS Software Protocol V.3

DOI
dx.doi.org/10.17504/protocols.io.3byl497wjgo5/v3

Leonardo Zilli[1], Erica Andreose[1], Salvatore Di Marzo[1]

[1]University of Bologna

Leonardo Zilli
University of Bologna

OPEN ACCESS

DOI: dx.doi.org/10.17504/protocols.io.3byl497wjgo5/v3

Protocol Citation: Leonardo Zilli, Erica Andreose, Salvatore Di Marzo 2024. IRIS Software Protocol. **protocols.io** **https://dx.doi.org/10.17504/protocols.io.3byl497wjgo5/v3**Version created by **Leonardo Zilli**

Protocol status: Working
We use this protocol and it's working

Created: April 12, 2024

Last Modified: May 23, 2024

Protocol Integer ID: 100182

Funders Acknowledgement:
University of Bologna
Grant ID:
https://ror.org/01111rn36

Abstract

We present a step-by-step methodology for tracking the coverage of the IRIS dataset in the OpenCitations Corpus. The methodology filters the original IRIS dataset and the OpenCitations Meta and Index dumps, producing two novel dataset containing the data that are then queried to answer the 5 research questions.

Guidelines

To allow complete reproducibility of the protocol, links to the data used are provided here.
Download link for the UNIBO IRIS bibliographic data dump, dated 14 March 2024:
https://amsacta.unibo.it/id/eprint/7608/
OpenCitations Meta CSV dataset of all bibliographic metadata, dated April 2024:
https://doi.org/10.6084/m9.figshare.21747461.v8

OpenCitations Index CSV dataset of all the citation data, dated November 2023:
https://doi.org/10.6084/m9.figshare.24356626.v2

The code used for this research, along with the data produced to answer to the research questions can be found in the **github repository** (**doi:10.5281/zenodo.11262417**)

Safety warnings

It is recommended to run the code on a machine that has at least 16gb of RAM memory available.
The results and timings presented in this protocol have been obtained with a 6 cores CPU.

Before start

Before starting, we suggest cloning the repository of the code by running the following command:

```
git clone git@github.com:open-sci/2023-2024-atreides-code.git
```

We also recommend, in case you would want to run the optional step of matching the id-less entities to Meta through the matching of the title, to create a *.env* file in the root of the folder and place your OpenCitations API key (you can obtain **here**) like so:

```
OC_APIKEY="<YOUR_API_KEY>"
```

We suggest to make sure you have Python3.x installed on your computer.
In order to correctly execute the provided scripts, you also must install the required libraries: **requirements.txt**. You can do so by running the following command:

```
pip install -r requirements.txt
```

IRIS Dataset Preparation

1 Make sure that the IRIS dump is present in the work directory.

1.1 Download the **iris dump** and place it in a 'data/' folder in your work directory. It is not required to unzip the archive.

Dataset	
UNIBO IRIS bibliographic data dump, dated 14 March 2024	NAME
https://amsacta.unibo.it/7608/1/iris-data-2024-03-14.zip	LINK

Create iris_in_meta dataset

2 This step will create a version of the OpenCitations Meta dump that is transformed and filtered according to the elements in the IRIS dump. This new dataset is stored in a parquet format that makes it lean and fast to query.

If you want to skip the creation of the dataset, you can download the final dataset **here**

Dataset	
Iris in Meta	NAME
https://doi.org/10.6084/m9.figshare.25879420	LINK

2.1 Download the **meta dump** and place it in the 'data/' folder in your work directory.

Dataset

OpenCitations Meta CSV dataset of all bibliographic metadata NAME

https://doi.org/10.6084/m9.figshare.21747461.v8 LINK

2.2 The purpose of this step is to read each CSV file in the Meta dump and process it by applying the following operations: 15m

1. Select the ['**id**', '**title**', '**type**'] columns
2. Extract from the '**id**' column the *omid*, and the *doi, isbn and pmid* if present through a regex pattern search. These 4 different elements are inserted into a new column created for each.
3. Create a new '**id**' column by combining the '**doi**', '**isbn**', and '**pmid**' columns, preferring the first non-null value.
4. Get rid of the '**doi**', '**isbn**', and '**pmid**' columns
5. Remove null values from the new **'id**' column
6. Perform an inner join with the *dois_isbns_pmids* dataframe

Note

The *dois_isbns_pmids* dataframe is created before the manipulation of the first Meta dump file.
It contains all ***valid*** DOI, ISBN and PMID present in the IRIS dump, along with their *iris_id* identifier.

7. Write the resulting dataframe to a .parquet file.

You can perform this step by using the following command:

Command

```
python3 meta_to_parquet.py -meta <path_to_meta_zip> -iris
<path_to_iris_zip>
```

Expected result

After the program has finished processing all the files, a '*iris_in_meta*' folder should have appeared in '*data/*'.

The dataset should have the following shape:

shape: (5, 5)

id	title	type	omid	iris_id
str	str	str	str	i32
"doi:10.1038/s41467-021-26551-x"	"Genome-wide Association Study Identifies Susceptibility Loci For Acut…	"journal article"	"omid:br/061401295311"	788512
"doi:10.1109/mmm.2016.2525119"	"Smart Solutions In Smart Spaces: Getting The Most From Far-Field Wire…	"journal article"	"omid:br/062602181087"	263669
"doi:10.1063/1.5007641"	"Results From CUORE And CUORE-0"	"proceedings article"	"omid:br/062502355373"	330526
"doi:10.1177/1362480619859350"	"Immigration Detention As Social Defence: Policing 'Dangerous Mobility…	"journal article"	"omid:br/062401318568"	441872
"doi:10.1016/j.ejso.2022.04.011"	"Trial Sequential Analysis Of Randomized Controlled Trials On Neoadjuv…	"journal article"	"omid:br/061201741385"	684086

2.3 It is also possible to attempt to retrieve and enrich the Iris in Meta dataset with the identifiers of the elements in the IRIS dump that do not have any DOI, ISBN, or PMID. This is done by querying the OpenCitations Meta SPARQL endpoint to search for each entitity by their *title.* From our tests this optional step was able to retrieve 150 additional entities.

3h

This is an optional step. This step has not been performed in the presented state of our research as its result can vary and it could lead to reproducibility incongruences.
We decided to report this only for completeness' sake.

Command

```
python3 meta_to_parquet.py -meta <path_to_meta_zip> -iris
<path_to_iris_zip> --search_for_titles
```

Safety information

WARNING: this will take around 3 hours to complete.

Create iris_in_index dataset

2h 30m

3 This step will create a version of the OpenCitations Index dump that is transformed and filtered according to the elements in the Iris in Meta dataset. This new dataset is also stored in a parquet format.

If you want to skip the creation of the dataset, you can download the final dataset **here**

Dataset	
Iris in Index	NAME
https://doi.org/10.6084/m9.figshare.25879441	LINK

3.1 Download the **index dump** and place it in the 'data/' folder in your work directory.

Dataset	
OpenCitations Index CSV dataset of all the citation data	NAME
https://doi.org/10.6084/m9.figshare.24356626.v2	LINK

3.2 The purpose of this step is to read all archives containing the CSV files of the Index dump and process it by applying the following operations: 2h 30m

1. Select the ['**id**', '**citing**', '**cited**'] columns
2. Filter the Dataframe to keep all rows that have, either in the **'cited'** or '**citing**' an id that is present in the *omids_list* Dataframe.

Note

The *omids_list* Dataframe contains a list of all the omid identifiers present in the Iris in Meta dataset.

3. Write each dataframe to a .parquet file.

You can perform this step by using the following command:

Command

```
python3 index_to_parquet.py --index_dump <path_to_index>
```

Safety information

WARNING: this will take around 1.5 hours to complete.

Expected result

After the program has finished processing all the files, a '*iris_in_index*' folder should have appeared in '*data/*'.

The dataset should have the following shape:

shape: (5, 3)

id	citing	cited
str	str	str
"oci:062102010944-06403538172"	"omid:br/062102010944"	"omid:br/06403538172"
"oci:062102010936-061901873468"	"omid:br/062102010936"	"omid:br/061901873468"
"oci:061302445980-0620582911"	"omid:br/061302445980"	"omid:br/0620582911"
"oci:062103078814-061903865447"	"omid:br/062103078814"	"omid:br/061903865447"
"oci:06403471424-06703722896"	"omid:br/06403471424"	"omid:br/06703722896"

Research Question answering

4 Each substep in this step will explain the answering process of each of the research questions mentioned in the abstract of this protocol.

You can decide to run the code for answering to a specific RQ by specifying it in the command used to run the script. It is also possible to answer all research questions at once by not specifying a specific one to the script, like so:

Command

```
python3 answer_research_questions.py
```

4.1 ***RQ1: What is the coverage of the publications available in IRIS, that strictly concern research conducted within the University of Bologna, in OpenCitations Meta?***

This research question is answered by simply computing the length of the Iris in Meta dataframe.

You can run the code to answer to this research question using the following command:

Command

```
python3 answer_research_questions.py -rq 1
```

Expected result

117764

4.2 ***RQ2: What are the types of publications that are better covered in the portion of OpenCitations Meta covered by IRIS?***

This research question is answered by grouping the Iris in Meta dataframe by the '**type'** column and then counting the length of the resulting dataframe.

You can run the code to answer to this research question using the following command:

Command

```
python3 answer_research_questions.py -rq 2
```

Expected result

```
| journal article       ¦ 104539 |
| proceedings article   ¦ 5608   |
| book chapter          ¦ 4482   |
| book                  ¦ 1482   |
| no type               ¦ 1450   |
| ...                   ¦ ...    |
| dataset               ¦ 6      |
| dissertation          ¦ 2      |
| series                ¦ 1      |
| computer program      ¦ 1      |
| book series           ¦ 1      |
```

4.3 ***RQ3: What is the amount of citations (according to OpenCitations Index) the IRIS publications included in OpenCitations Meta are involved in (as citing entity and as cited entity)?***

This research question is answered by simply computing the length of the Iris in Index dataframe.

You can run the code to answer to this research question using the following command:

Command

```
python3 answer_research_questions.py -rq 3
```

Expected result

7859226

4.4 *RQ4: How many of these citations come from and go to publications that are not included in IRIS?*

This research question is answered by filtering each '**citing**' and '**cited**' column of the Iris in Index dataset to remove all rows in which elements from the aforementioned *omids_list* are present. The length of each of the two resulting dataframes are then computed to get the final answer.

You can run the code to answer to this research question using the following command:

Command

```
python3 answer_research_questions.py -rq 4
```

Expected result

citing	cited
3562668	3950823

4.5 *RQ5: How many of these citations involve publications in IRIS as both citing and cited entities?*

This research question is answered by filtering the Iris in Index dataset to keep only the rows in which elements from the aforementioned *omids_list* are present in either the '**citing**' or in the '**cited**' columns. The length of the resulting dataframe is then computed to get the final answer.

You can run the code to answer to this research question using the following command:

Command

```
python3 answer_research_questions.py -rq 5
```

Expected result

358877