Sep 20, 2024 Version 5

Exploring the Coverage of the Scientific Production of the University of Bologna in OpenCitations V.5

DOI
dx.doi.org/10.17504/protocols.io.3byl497wjgo5/v5

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

[1]University of Bologna

Open Science Course - Te…

Leonardo Zilli
University of Bologna

DOI: dx.doi.org/10.17504/protocols.io.3byl497wjgo5/v5

Protocol Citation: Leonardo Zilli, Erica Andreose, Salvatore Di Marzo 2024. Exploring the Coverage of the Scientific Production of the University of Bologna in OpenCitations. **protocols.io** **https://dx.doi.org/10.17504/protocols.io.3byl497wjgo5/v5**Version created by **Leonardo Zilli**

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

Created: April 12, 2024

Last Modified: September 20, 2024

Protocol Integer ID: 106942

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

Abstract

We present a step-by-step methodology for tracking the coverage of publications available in the CRIS (Current Research Information System) infrastructure of the University of Bologna and implemented in the IRIS platform, within the databases of OpenCitations.

The methodology filters and transforms data dumps from IRIS and OpenCitations Meta and Index to create novel datasets that are used to perform quantitative analysis on the subject matter. Specifically, we quantify the proportion of IRIS publications included in OpenCitations Meta, examine the types of publications best covered, evaluate the number of citations within OpenCitations Index involving IRIS publications as either citing or cited entities, and the extent to which these citations reference works outside of IRIS.

The research questions addressed in the study are:

1. What is the coverage of the publications available in IRIS (strictly concerning research conducted within the University of Bologna) in OpenCitations Meta?
2. What are the types of publications that are better covered in the portion of OpenCitations Meta covered by IRIS?
3. What is the amount of citations (according to OpenCitations Index) coming from the IRIS publications that is involved in OpenCitations Meta (as citing entity and as cited entity)?
4. How many of these citations come from and go to publications that are not included in IRIS?
5. How many of these citations involve publications in IRIS as both citing and cited entities?

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 4 June 2024:
https://amsacta.unibo.it/id/eprint/7736/

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.11262416**)

The tables and values contained in the "expected result" snippets in this protocol are to be intended as reduced exemplars of the output expected from each step. The actual data may change depending on the version of the datasets used (e.g. a new version of Meta).

Safety warnings

It is recommended to run the code on a machine with at least 16gb of RAM memory available.

Before start

Note: Python 3.11.5 or previous versions are required to run the software.

Prepare the working environment by executing the following commands:

```
# Clone the repository
git clone https://github.com/open-sci/2023-2024-atreides-code

# Move to the repository folder
cd 2023-2024-atreides-code

# Install the package and dependencies
pip install .
```

In case you would want to run *optional* step 5.2 to match the id-less entities, create a .env file in the root folder of the software and store your OpenCitations API key (which you can obtain **here**) in it like so:

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

Data preparation

1 The first data dump used in the research comes from the IRIS infrastructure of the University of Bologna.

Dataset

UNIBO IRIS bibliographic data dump, dated 4 June 2024 NAME

https://doi.org/10.6092/unibo/amsacta/7736 LINK

It comprises seven CSV files that describe 304,983 bibliographic entities and has a total size of 267 MB.
Each CSV file contains metadata regarding specific aspects of the bibliographic entities. The files are the following:

- "ODS_L1_IR_ITEM_CON_PERSON.csv": information about the people involved in the publications (authors, editors, etc.)
- "ODS_L1_IR_ITEM_DESCRIPTION.csv": the string of the authors and other related metadata of publications
- "ODS_L1_IR_ITEM_IDENTIFIER.csv": the identifiers (including DOIs) of publications
- "ODS_L1_IR_ITEM_LANGUAGE.csv": the language in which the publication has been written (when applicable)
- "ODS_L1_IR_ITEM_MASTER_ALL.csv": basic metadata information of publications (title, date of publication, type of publication)
- "ODS_L1_IR_ITEM_PUBLISHER.csv": the publishers of publications
- "ODS_L1_IR_ITEM_RELATION.csv": additional metadata related to the context of publications (publication venue, editors, etc.)

The dataset is downloaded from **here** and placed in the 'data/' folder of our working directory. Unzipping the archive is not required.

2 The second data dump used in the research comes from OpenCitations Meta.

Dataset

OpenCitations Meta CSV dataset of all bibliographic metadata NAME

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

This dataset contains all the bibliographic metadata (in CSV format) included in OpenCitations Meta. In particular, each line of the CSV file defines a bibliographic resource, and includes the following information:

- **[field "id"]** the IDs for the document described within the line;
- **[field "title"]** the document's title;
- **[field "author"]** the authors of the document;
- **[field "pub_date"]** the date of publication;
- **[field "venue"]** information about the venue, i.e. the bibliographical resource to which the document belongs;
- **[field "volume"]** the volume sequence identifier (e.g. a number) to which the entity belongs;
- **[field "issue"]** the issuesequence identifier (e.g. a number) to which the entity belongs;
- **[field "page"]** the page range of the resource described in the row;
- **[field "type"]** the type of resource described in the row;
- **[field "publisher"]** the entity responsible for making the resource available;
- **[field "editor"]** the editors of the document.

This version of the dataset contains:

- 114,703,611 bibliographic entities
- 298,847,794 authors and 2,465,711 editors
- 711,711 publication venues
- 241,783 publishers

The zipped dataset weighs 11 GB, while, when extracted, it weighs 46 GB on an ext4 filesystem. Additional information about OpenCitations Meta at **official webpage**.

The dataset is downloaded from **here** and placed it in the 'data/' folder of the software's directory. Unzipping the archive is not required.

3 The third data dump used in the research comes from the OpenCitations Index.

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

This dataset contains all the citation data (in CSV format) included in the **OpenCitation Index**, released on November 29, 2023. In particular, each line of the CSV file defines a citation, and includes the following information:

- **[field "oci"]** the Open Citation Identifier (OCI) for the citation;

- **[field "citing"]** the OMID of the citing entity;
- **[field "cited"]** the OMID of the cited entity;
- **[field "creation"]** the creation date of the citation (i.e. the publication date of the citing entity);
- **[field "timespan"]** the time span of the citation (i.e. the interval between the publication date of the cited entity and the publication date of the citing entity);
- **[field "journal_sc"]** it records whether the citation is a journal self-citations (i.e. the citing and the cited entities are published in the same journal);
- **[field "author_sc"]** it records whether the citation is an author self-citation (i.e. the citing and the cited entities have at least one author in common).

The information for each citation is sourced from OpenCitations Meta.

This version of the dataset contains:
- 1,975,552,846 citations;
- 89,920,081 bibliographic resources.

The size of the zipped archive is 26.8 GB, while the size of the unzipped CSV file is 171 GB.

Creation of a unique list of PIDs

4 This section will create a list list of unique identifiers from the IRIS dump, which we use to search for BRs within OC Meta. Specifically, we extract all entities in IRIS with DOI, ISBN, or PMID identifiers, as these are the only persistent IDs (PID) that are present both in IRIS and OCMeta

4.1 We first read the "ODS_L1_IR_ITEM_MASTER_ALL.csv" and "ODS_L1_IR_ITEM_MASTER_ALL.csv" files of the IRIS dataset and we join them by the values **ITEM_ID** column.

We then filter the resulting dataframe to keep only the entries that have at least a non-null DOI, ISBN or PMID.
We also keep the **OWNING_COLLECTION** column to denote the labels of the types of the entries.

Expected result

ITEM_ID	IDE_DOI	IDE_ISBN	IDE_PMID	OWNING_COLLECTION
156671	"10.1688/9783866187337"	"9783866186330; 9783866187337"	null	41
9754	null	"9788867414727"	null	57

From this filtering process the first of the datasets produced by the research is created, ***Iris No ID***, containing the metadata of the IRIS records without a DOI, ISBN, or PMI.
The script to execute to create this dataset is the following:

Command

```
python3 scripts/create_datasets.py -meta <path_to_meta_zip> -iris
<path_to_iris_zip> --iris_no_ids
```

Note

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

Dataset

Iris No ID NAME

https://doi.org/10.6084/m9.figshare.25897759.v2 LINK

4.2 We extract a list of all the PIDs for each entry of the filtered IRIS dataframe by extracting the values of the columns **IDE_DOI**, **IDE_ISBN**, and **IDE_PMID**. Malformed identifiers are sanitized and invalid ones discarded, and all identifiers are unified in format and converted to lowercase for consistency with the Meta dataset.

Finally, they are all stored in a single list that we'll use to filter the Meta dump.

Expected result

iris_id	iris_type	id
156671	41	"doi:10.1688/9783866187337"
148354	35	"doi:10.1007/s00180-012-0319-z"
146851	35	"doi:10.1002/cmdc.201100471"
147819	35	"doi:10.1097/gme.0b013e318240fe3d"
148141	57	"doi:10.1109/aero.2012.6187311"

4.3 In order to remove the duplicates present in the list, we first filter this list to keep only the first occurrence of cases in which the same IRIS record had more than one PID associated to it. Given how the list has been constructed (DOIs are listed before PMIDs and ISBNs are last), the filtering is applied with a hierarchical order of preference for the picking of a single PID for each IRIS entry.

4.4 Next, multiple IRIS records associated to the same PID are deduplicated, keeping only one occurrence of each PID. To pick the best entry out of the records that shared the same PID, the three types of identifier are processed separately.

To deduplicate DOIs, the following methodology is implemented:

1. Out of the duplicates with the same DOI and the same type, the first one occurring in the list is kept
2. For the duplicates with the same DOI and different type, the duplicates are sorted according to a priority list based on the type of the record, and the resulting top record is kept.

The preference list for DOIs is devised as follows, from highest priority to lowest:
1. 35 - 1.01 Articolo in rivista
2. 50 - 3.02 Curatela
3. 41 - 2.01 Capitolo / saggio in libro
4. 57 - 4.01 Contributo in Atti di convegno

4.5 To deduplicate PMIDs, first the duplicates with type '40 - Abstract' are removed and the first occurrence of each PMID with type '35 - Articolo in rivista' are kept.

4.6 To deduplicate ISBNs, the same methodology used for the DOIs, with a different priority list, is implemented:

1. Out of the duplicates with the same ISBN and the same type, the first one occurring in the list is kept
2. For the duplicates with the same ISBN and different type, the duplicates are sorted according to a priority list based on the type of the record, and the resulting top record is kept.

The preference list for ISBNs is devised as follows, from highest priority to lowest:
1. 49 - 3.01 Monografia
2. 35 - 1.01 Articolo in rivista

Creation of the *Iris In Meta* dataset

5 Now that we have a clean, deduplicated list of external ids, we can filter the OCMeta dump to keep only the entries that have a PID that appears in the list.

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.

Note

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

Dataset

Iris in Meta NAME

https://doi.org/10.6084/m9.figshare.25879420.v2 LINK

5.1 Each CSV file in the Meta dump is transformed 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
7. Write the resulting dataframe to a .parquet file.

The resulting dataset will be composed of thousands of .parquet files, each corresponding to a CSV file of the OCMeta dump.
The version of the OCMeta dump used in this study is affected by cases of duplicates entries where the same BR appears multiple times with different OMIDs. To remove these duplicates, a $priority$ dataframe is created, pictured below:

Meta type	IRIS type	priority
Journal articles	35 - 1.01 Articolo in rivista	0
Book chapters	41 - 2.01 Capitolo / saggio in libro	1
Book chapters	42 - 2.02 Prefazione	2

The dataset is filtered like so:

1. The whole dataset is read
2. A left join is performed between the dataset and the $priority$ dataframe
3. The joined dataset is sorted by the **priority** column
4. The entries are grouped by their **id**
5. The first entry for each group is selected
6. The **priority** column is dropped
7. The **type_label** values are replaced with the actual label from IRIS
8. The dataset is saved to a single .parquet file

You can perform this step by executing the following command:

Command

```
python3 scripts/create_datasets.py -meta <path_to_meta_zip> -iris
<path_to_iris_zip> --iris_in_meta
```

Expected result

After the program has finished processing all the files, a '*iris_in_meta*' folder should have appeared in '*data/*'. Inside, of this folder is located the 'iris_in_meta.parquet' file containing the processed dataset.

The dataset has the following shape:

	id	title	meta_type	omid	iris_id	iris_type
	"doi:10.1007/s10334-004-0090-4"	"Versatile Coil Design And Positioning Of Transverse-Field RF Surface Coils For Clinical 1.5-T MRI Ap...	"journal article"	"omid:br/06101045684"	64745	"1.01 Articolo in rivista"
	"doi:10.1136/archdischild-2017-314663"	"Tricky Case Of Takayasu Arteritis In A Young Child Presenting With Heart Failure And Femoral Pulses"	"journal article"	"omid:br/061402023621"	349405	"1.01 Articolo in rivista"

5.2 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 scripts/create_datasets.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.

Creation of the *Iris Not In Meta* dataset

6 The optional dataset *Iris Not In Meta* can be created after *Iris In Meta.*
This dataset contains all the bibliographic records from the IRIS dump that have a DOI, ISBN or PMID but that have not been found in the OCMeta dump.

Note

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

Dataset

Iris Not In Meta	NAME
https://doi.org/10.6084/m9.figshare.25897708.v2	LINK

6.1 To create this dataset, the following methodology is adopted:

1. Retrieve the deduplicated list of unique PIDs from the IRIS dump as in section 4.

2. Read the *Iris In Meta* dataset.
3. Perform an anti join between the *Iris In Meta* dataframe and the list of PIDs on the **'iris_id'** column.
4. Write the resulting dataframe to a .parquet file.

You can perform this step by executing the following command:

Command

```
python3 scripts/create_datasets.py -meta <path_to_meta_zip> -iris
<path_to_iris_zip> --iris_not_in_meta
```

Creation of the *Iris In Index* dataset

1h 30m

7 This step will create a version of the OpenCitations Index dump that is filtered to keep only the records that are present in the *Iris In Meta* dataset. This is done by first extracting all the OMIDs from *Iris in Meta* dataset and filtering the OCIndex CSV files to retrieve all citations in which an OMID from the list appears as either '**cited**' or as '**citing**' entity.
This new dataset is also stored in the parquet format.

Note

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.v2	LINK

7.1 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. Extract the list of OMIDs $omids\_list$ from the *Iris In Meta* dataset by converting the '**omid**' column to a list.
2. Read the OC Index dump, selecting the ['**id**', '**citing**', '**cited**'] columns only.
3. Filter the Index Dataframe to keep all rows that have, either in the **'cited'** or '**citing**' columns an OMID that is present in the $omids\_list$.
4. Write each dataframe to a .parquet file.

You can perform this step by using the following command:

Command

```
python3 scripts/create_datasets.py -meta <path_to_meta_zip> -iris
<path_to_iris_zip> -index <path_to_index_zip> --iris_in_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 has the following shape:

id	citing	cited
"oci:06101850106-06201834752"	"omid:br/06101850106"	"omid:br/06201834752"
"oci:06101850106-06201834594"	"omid:br/06101850106"	"omid:br/06201834594"
"oci:06101850106-06301833659"	"omid:br/06101850106"	"omid:br/06301833659"
"oci:06101850106-06301833261"	"omid:br/06101850106"	"omid:br/06301833261"
"oci:06101850101-06301833742"	"omid:br/06101850101"	"omid:br/06301833742"

Research Question answering

8 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 a specific RQ by specifying its number using the -*rq* argument of 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
```

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

The methodology used to answer this research question is the following:

1. Read the *Iris In Meta* dataset.

2. Count the number of rows of the *Iris in Meta* dataset.

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

Command

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

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

The methodology used to answer this research question is the following:
1. Read the *Iris In Meta* dataset.
2. Group the *Iris In Meta* dataset by the **'type'** column.
3. Count the number of rows in each group.

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

Command

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

8.3 ***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)?***

The methodology used to answer this research question is the following:
1. Read the *Iris In Index* dataset.
2. Count the number of rows of the *Iris in Index* dataset.

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

Command

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

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

The methodology used to answer this research question is the following:

1. Read the *Iris In Meta* dataset.
2. Extract a list of OMIDs from *Iris In Meta* by converting the **'omid'** column to a list.
3. Read the *Iris In Index* dataset.
4. Filter the **'citing'** column of *Iris In Index* to keep only rows in which the value of the column is contained in the list of OMIDs.
5. Filter the **'cited'** column of *Iris In Index* to keep only rows in which the value of the column is contained in the list of OMIDs.
6. Count the rows of each of the two dataframes resulting from the previous steps and concatenate the two values to a single dataframe.

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

Command

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

8.5 ***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.

The methodology used to answer this research question is the following:

1. Read the *Iris In Meta* dataset.
2. Extract a list of OMIDs from *Iris In Meta* by converting the **'omid'** column to a list.
3. Read the *Iris In Index* dataset.
4. Filter the **'citing'** and **'cited'** columns of *Iris In Index* to keep only rows in which the value of both columns are contained in the list of OMIDs.
5. Count the rows of the dataframe resulting from the previous step.

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

Command

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