Aug 07, 2025 Version 6

Analysing the coverage of the University of Bologna's bibliographic and citation metadata in OpenCitations collections V.6

DOI
https://dx.doi.org/10.17504/protocols.io.g6xmbzfk7

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

[1]University of Bologna

Leonardo Zilli
University of Bologna

DOI: https://dx.doi.org/10.17504/protocols.io.g6xmbzfk7

Protocol Citation: Leonardo Zilli, Erica Andreose, Salvatore Di Marzo 2025. Analysing the coverage of the University of Bologna's bibliographic and citation metadata in OpenCitations collections. **protocols.io** https://dx.doi.org/10.17504/protocols.io.g6xmbzfk7Version created by Leonardo Zilli

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

Created: August 03, 2025

Last Modified: August 07, 2025

Protocol Integer ID: 223949

Keywords: publications in iris, proportion of iris publication, involving iris publication, iris publication, coverage of the publication, coverage of publication, publication, citation, university of bologna, many of these citation, number of citation, amount of citation, cited entity, current research information system, types of publication, concerning research, databases of opencitation, research, citations reference, opencitations index, infrastructure of the university, coverage of the scientific production, opencitations meta, research question, portion of opencitations meta, data dumps from iris, citing entity, available in iris, database, iris, iris platform, university, quantitative analysis on the subject matter, bologna, citation metadata in opencitations collection, citation metadata, coverage of the university, opencitations collection

Funders Acknowledgements:
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 and additional material used in the study are provided here.
Download link for the UNIBO IRIS bibliographic data dump, dated 30 May 2025, updated on 3 July 2025:
https://doi.org/10.6092/unibo/amsacta/8427

OpenCitations Meta CSV dataset of all bibliographic metadata (June 2025):
https://doi.org/10.5281/zenodo.15625651

OpenCitations Index CSV dataset of all the citation data (July 2025):
https://doi.org/10.6084/m9.figshare.24356626.v6

The output datasets produced to answer to the research questions are made freely available on FigShare:
Iris in Meta – Published on August 7, 2025 (**10.6084/m9.figshare.25879420.v3**)
Iris in Index – Published on August 7, 2025 (**10.6084/m9.figshare.25879441.v3**)
Iris Not in Meta – Published on August 7, 2025 (**10.6084/m9.figshare.25897708.v3**)
Iris No ID – Published on August 7, 2025 (**10.6084/m9.figshare.25897759.v3**)

All code used for data processing and analysis for the current study is openly available on **GitHub** and archived on Zenodo (**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: Software developed and tested using Python 3.12

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 required dependencies using uv
# uv install options: https://docs.astral.sh/uv/getting-started/installation/
uv sync

# Activate the virtual environment
source .venv/bin/activate
```

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 gathering

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 30 May 2025 NAME

https://doi.org/10.6092/unibo/amsacta/8427 LINK

It comprises seven CSV files that describe 402,505 bibliographic entities and has a total size of 430 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/10.5281/zenodo.15625651 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:

- 124,526,660 bibliographic entities
- 376,295,095 authors, 2,765,927 editors, and 103,928,927 publishers (counted by their roles, without disambiguating individual entities)
- 1,019,563 publication venues

The compressed dataset weighs 12G, while, when extracted, it weighs 49G 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.v6 LINK

This dataset contains all the citation data (in CSV format) included in the OpenCitation Index (https://opencitations.net/index), released on July 10, 2025. 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).

Note: the information for each citation is sourced from **OpenCitations Meta**, a database that stores and delivers bibliographic metadata for all bibliographic resources included in the OpenCitations Index. The data provided in this dump is therefore based on the state of OpenCitations Meta at the time this collection was generated.

This version of the dataset contains:

- 2,216,426,689 citations

The size of the zipped archive is 38.8 GB, while the size of the unzipped CSV file is 242 GB.

Creation of a list of unique PIDs

4 This stage collects all bibliographic entities from IRIS and processes them to create a clean, deduplicated list of unique identifiers (PIDs), 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 types of PIDs that are present both in IRIS and OC Meta

4.1 We first read the "ODS_L1_IR_ITEM_MASTER_ALL.csv" and "ODS_L1_IR_ITEM_IDENTIFIER.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 records.

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, PMID, or ISBN. The script to execute to create this dataset is the following:

Command

new command name

```
python3 -m scripts.create_datasets -meta <path_to_meta_zip> -iris <path_to_iris_zip> --iris_no_id
```

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.v3	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. The following deduplication methodology is implemented:

1. Within groups sharing the same type and PID, the record with the most complete metadata (fewest null values) is kept.
2. For groups sharing the same PID but different types, records are sorted according to a priority system, and the top record is kept. The priority tables are as follows:

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

The preference list for PMIDs is devised as follows, from highest priority to lowest:
1. 35 - 1.01 Articolo in rivista

The preference list for ISBNs is devised as follows, from highest priority to lowest:
1. 50 - 3.02 Curatela
2. 49 - 3.01 Monografia
3. 35 - 1.01 Articolo in rivista

By the end of this process, a list of unique BRs remains, which will be used for for the mapping with OpenCitations Meta.

Creation of the *Iris in Meta* dataset

5 The **Comparator** stage, uses the clean list of unique PIDs to filter the OC Meta dump in order to create a version of OC Meta that is transformed and filtered according to the elements in the IRIS dump.

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.v3	LINK

5.1 Each CSV file in the Meta dump is transformed by applying the following operations: 15m

1. Select the ['**id**', '**title**', '**type**', '**pub_date**'] 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 final dataset is then filtered like so:

1. All .parquet files are read into a single dataframe.
2. A left join is performed between the dataframe and the $priority$ dataframe.
3. The joined dataframe is sorted by the **priority** column first and **pub_date** column after, with null values placed last.
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 **pub_date** column is used to filter out the records published after 2025. In cases where the pub_date from meta is not available, we use the DATE_ISSUED_YEAR field from the IRIS dump to do this filtering.
9. The dataset is saved to a single .parquet file and intermediate .parquet files are deleted.

You can create the *Iris in Meta* dataset by executing the following command:

Command

new command name

```
python3 -m scripts.create_datasets -meta <path_to_meta_zip> -iris
<path_to_iris_zip> --iris_in_meta -yc 2025
```

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	pub_date	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"	2013	"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"	2010-10-23	"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

new command name

```
python3 scripts/create_datasets -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.v3	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

new command name

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

Creation of the *Iris in Index* dataset

40m

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.v3	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: 40m

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**', '**creation**'] 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.

The final dataset is then filtered like so:

1. All .parquet files are read into a single dataframe.
2. The dataframe is sorted by the **id** column first and **creation** column after, with null values placed last.
3. The '**creation**' column is used to filter out the records published after 2025.
4. Records are deduplicated by retaining only the first occurrence of each duplicated row by checking the '**id**' column.
5. The whole dataset is saved to a single .parquet file and all intermediate files are deleted.

You can create the *Iris in Index* dataset by using the following command:

Command

new command name

```
python3 -m scripts.create_datasets -meta <path_to_meta_zip> -iris
<path_to_iris_zip> -index <path_to_index_zip> --iris_in_index -yc 2025
```

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	creation
	"oci:06101850106-06201834752"	"omid:br/06101850106"	"omid:br/06201834752"	2018-07-20
	"oci:06101850106-06201834594"	"omid:br/06101850106"	"omid:br/06201834594"	2018-07
	"oci:06101850106-06301833659"	"omid:br/06101850106"	"omid:br/06301833659"	2019-05-10

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

new command name

```
python3 -m scripts.answer_research_questions
```

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

new command name

```
python3 -m scripts.answer_research_questions -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

new command name

```
python3 -m scripts.answer_research_questions -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

new command name

```
python3 -m scripts.answer_research_questions -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 dataframes resulting from the previous steps.

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

Command

new command name

```
python3 -m scripts.answer_research_questions -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

new command name

```
python3 -m scripts.answer_research_questions -rq 5
```