Research on ERIH PLUS approved SSH journals present in OpenCitations Meta database V.3

Ali Ghasempouri[1], maddalena.ghiotto[1], sebastiano.giacomini[1]

[1]unibo

maddalena.ghiotto

VERSION 3

MAY 29, 2023

OPEN ACCESS

DOI: dx.doi.org/10.17504/protocols.io.5jyl8jo1rg2w/v3

Protocol Citation: Ali Ghasempouri, maddalena.ghiotto, sebastiano.giacomini 2023. Research on ERIH PLUS approved SSH journals present in OpenCitations Meta database . **protocols.io** https://dx.doi.org/10.17504/protocols.io.5jyl8jo1rg2w/v3

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

Created: May 26, 2023

Last Modified: May 29, 2023

DISCLAIMER

DISCLAIMER – FOR INFORMATIONAL PURPOSES ONLY; USE AT YOUR OWN RISK

The protocol content here is for informational purposes only and does not constitute legal, medical, clinical, or safety advice, or otherwise; content added to protocols.io is not peer reviewed and may not have undergone a formal approval of any kind. Information presented in this protocol should not substitute for independent professional judgment, advice, diagnosis, or treatment. Any action you take or refrain from taking using or relying upon the information presented here is strictly at your own risk. You agree that neither the Company nor any of the authors, contributors, administrators, or anyone else associated with protocols.io, can be held responsible for your use of the information contained in or linked to this protocol or any of our Sites/Apps and Services.

PROTOCOL integer ID:
82530

ABSTRACT

In this study, we present a comprehensive workflow to assess the coverage of publications in Social Science and Humanities (SSH) journals indexed in ERIH-PLUS and their Open Access status according to the Directory of Open Access Journals (DOAJ). The workflow utilizes three data sources: ERIH-PLUS, OpenCitations Meta, and DOAJ.

The application of this workflow results in a dataset containing detailed information on SSH publications, including their disciplines, countries of origin, and Open Access status. Each step of the methodology enriches the dataset with new variables and insights. The output of this workflow includes discipline and country rankings, as well as visualizations to effectively communicate the findings. By following this step-by-step approach, researchers can better understand the landscape of SSH publications, identify trends in disciplines and countries, and evaluate the prevalence of Open Access in the field.

The software implemented by means of this methodology and related documentation can be found in our github repository that will eventually be published on Zenodo.

Retrieve OpenCitation Meta publication and Journals that a...

1 Starting from the ERIH-PLUS index of Social Science and Humanities approved journals dataset 📎 ERIHPLUSapprovedJournals.csv (downloaded 27/04/2023) we want to retrieve all the publications belonging to one of those journals, included in OpenCitations Meta database (https://opencitations.net/meta#:~:text=For%20each%20publication%2C%20the%20metadata,and%20PubMed%20Identifiers%20(PMIDs).)

1.1 In order to fulfill this task, we download the OpenCitations Meta data dump.
To process the files we run the function *process_files()* : this script processes CSV files in parallel.
This is how it works:

```
with concurrent.futures.ProcessPoolExecutor(max_workers=4) as
executor:
```

This line creates a ProcessPoolExecutor with 4 worker processes. The with statement is used to ensure that the executor is properly closed once the processing is done. The number 4 represents the maximum number of worker processes that will be created to execute the tasks concurrently. You can adjust this number based on the resources available on your system.

```
results = executor.map(process_file_wrapper, [(f, erih_plus_df)
for f in batch_files])
```

The executor.map() function is used to apply the process_file_wrapper function to a list of input arguments. In this case, the input arguments are tuples, each containing a file from the batch_files and the erih_plus_df DataFrame. The executor.map() function returns an iterator that yields the results of applying the process_file_wrapper function to each tuple in the list.

```
all_results.extend(results)
```

This line extends the all_results list with the results obtained from processing the current batch of files. The extend() method is used to add multiple items to the list at once.

1.2 Processing the ERIH-PLUS Journals dataset and matching venues identifiers with OpenCitations Meta dump's venues.
For this step, the function *process_meta_csv* () is called. It takes in input as parameters
1. the chunk of the csv that is being processed
2. and the ERIH-PLUS dataset as a dataframe.

Note

Input: ERIH-PLUS approved journal's dataset
Structured as follow:

	Journal ID	Print ISSN	Online ISSN	Original Title	International Title	Country of Publication	E
	486254	1989-3477	NaN	@tic.revista d'innovació educativa	@tic.revista d'innovació educativa	Spain	

OpenCitations Meta data about **venues** (issn that we need to decide how to retrieve)

Note

output: A dataset mapping OpenCitations Meta venue data (OMID and ISSN) to ERIH-PLUS venue data (Journal ID and ISSN).
This dataset will have the following structure:

	OC_omid	issn	EP_id	Publications_in_venue
	meta:br/060156	[4522-4592, 5687-3452]	503890	12
	meta:br/060164	isbn:242352513	783726	13526
	meta:br/060167	issn:4522-4592	503890	56

1.3 Processing whether each journal in the merged_data dataframe is Open Access based on its ISSN being present in the DOAJ data.
For this step, the ancillary function *process_doaj_file()* is called.
This function takes in input
1. doaj_df
2. merged_data
and returns a Dataframe structured as follows:

	OC_omid	issn	EP_id	Publications_in_venue	Open_Access
	meta:br/060167	[2423-5251, 8763-2891]	876390	12	Unknown
	meta:br/060167	[4522-4592, 5687-3452]	503890	56	True

The function follows these steps:

It first selects three columns from the DOAJ dataframe: the 5th, 6th, and 10th columns. The columns correspond to print ISSN, online ISSN, and Country of publisher (about the latter: see *step 2.3* below).

It then creates an empty dictionary, **open_access_dict**, which will map both print and online ISSNs to a Boolean value True that indicates Open Access status.

The function loops over the rows of the DOAJ dataframe, adding each print and online ISSN to the dictionary and setting its value to True.

It creates a list, **open_access_keys**, of all the keys (ISSNs) in the **open_access_dict**.

It then adds a new column, **'Open Access'**, to the **merged_data** dataframe and initializes all its values to **'Unknown'**.

The function goes over each ISSN in the **merged_data** dataframe. If this ISSN is found in the **open_access_keys**, the Open Access status of that row in **merged_data** is updated to True.

2

Retrieve data about countries and disciplines

2.1 Our second and third research question are
1. What are the disciplines that have more publications?
2. What are countries providing the largest number of publications and journals?

These information are both present in the ERIH-PLUS dataset.

the script '**Disciplines_Countries_classes.py'** allows for processing metadata from multiple sources to retrieve and count information about journals by country and discipline. The code is object-oriented, utilizing classes and inheritance for shared attributes and methods. Here's a breakdown of what each class does:
class ResultsProcessor: This is a base class that initializes some common attributes that are used in the child classes. The '**__init__**' method takes two parameters: '**meta_coverage**' and '**meta_coverage_processed_files**'. It reads the processed files into a dataframe and gets dataframes for ERIH PLUS and DOAJ data.

Disciplines:
This class also extends 'ResultsProcessor' and is used to process discipline information. The 'create_disciplines_dict' method creates a dictionary mapping each discipline to a list of journal IDs associated with that discipline.

As we can see every journal can have multiple disciplines:
In order to count the disciplines and understand which one has the highest number of publications we will need to disassemble them and map them individually with the venue.

2.2 **Countries:**
This class extends '**ResultsProcessor**' and is used to process country information. Its '**__init__**' method additionally selects certain columns from the DOAJ dataframe. The method '**create_countries_dict**' creates a dictionary mapping each country to a list of journal IDs that are published in that country. The method uses a helper function '**retrieve_doaj_country**' to retrieve country information for unmatched journals from the DOAJ dataset.

2.3 Checking wether missing countries in erih-plus are present in doaj dataset, where **Country of Publisher** contains the same country information as ERIH-PLUS **Country of Publication.**

1. It starts by creating a dataframe **'no_country_df'** that only includes rows from **'result_df'** where the '**EP_id**' is in the list of unmatched ISSNs.
2. It then cleans up the '**issn**' column in **'no_country_df'** by replacing the string representation of a list with an actual list of ISSNs.
3. It uses the explode function to transform each element of the 'issn' list to a row, replicating the index values.
4. It then merges **'no_country_df'** with **'doaj_df'** twice: first on '**Journal ISSN (print version)**', and then on '**Journal EISSN (online version)**', to try to match the ISSNs in both versions.
5. It cleans up the resulting dataframe by selecting only necessary columns, merging the country information from both '**Country of publisher_x**' and '**Country of publisher_y**' into a new '**Country**' column, and dropping duplicates.
6. It then separates the entries where '**Country**' is still NaN after the merge, storing these in **'unmatched_df'**.
7. It updates the **'countr_dict'** dictionary by adding the '**EP_id**' of each journal to the list corresponding to its '**Country**' in the dictionary.
8. Lastly, it checks for and handles some specific cases where the country names may appear differently but refer to the same country (e.g., 'Turkey' and 'Türkiye', 'Venezuela' and 'Venezuela, Bolivarian Republic of', 'Republic of').

Final counts

3 To answer our research questions, we will compute the **counts of publications and journals** for both disciplines and countries using the method ***counts()*** of the **CountsProcessor** class. This method will take in input either the country or the discipline **dictionary** created in the previous step (**2.1, 2.3**) and the **label** to set as the first column value in the final **dataset**.

The method iterates over the dictionary keys to filter the first output DataFrame (*SSH_Publications_in_OC_Meta_and_Open_Access_status.csv*) according to journals in the list specified as value, then it stores the length of the filtered DataFrame as the **count of the journals**. Lastly, it sums all the values in the column Publications_in_venue to calculate the count of publications.

About our research question "How many of the SSH journals are available in Open Access according to the data in DOAJ?": we will simply count the rows that have a "**TRUE**" value in the **Open Access** column of *SSH_Publications_in_OC_Meta_and_Open_Access_status.csv*

Visualize results

4 For each visualization, we use Python libraries like Matplotlib, Seaborn, Plotly to create the bar and pie charts and map, as follow:

- a bar plot that visualizes the number of publications for different disciplines. Each discipline is represented by a bar, and the height of the bar corresponds to the publication count. The plot also includes axis labels, a title, and gridlines on the x-axis. The colors used for the bars are defined in the colors list.
- a horizontal bar chart that shows the top 30 countries ranked by their journal count. Each country is represented by a bar, and the length of the bar corresponds to the journal count.
- a horizontal bar chart that shows the last 30 countries ranked by their journal count. Each country is represented by a bar, and the length of the bar corresponds to the journal count.
- a horizontal bar chart that shows the top 30 countries ranked by their publication count. Each country is represented by a bar, and the length of the bar corresponds to the publication count.
- a horizontal bar chart that shows the last 30 countries ranked by their publication count. Each country is represented by a bar, and the length of the bar corresponds to the publication count.
- a pie chart that illustrates the percentage of ERIH Plus journals that are covered in OpenCitations Meta and the percentage that are not covered. The chart includes labels for each section showing the coverage status and the corresponding percentage.
- a pie chart that represents the distribution of the 'Open Access' categories. Each category is represented by a slice of the pie, and the size of each slice corresponds to the count or percentage of occurrences.
- a choropleth representation of the publications by country. Each country is filled with a color based on its publication count, as specified by the 'Publication_count' column. Hovering over a country shows its name and additional information.