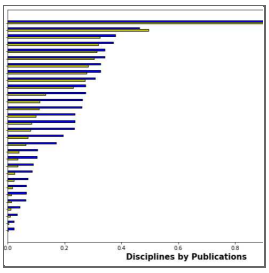


VERSION 5
AUG 16, 2023

OPEN ACCESS

DOI:
dx.doi.org/10.17504/protocols.io.5jyl8jo1rg2w/v5

Protocol Citation: Ali Ghasempouri, maddalena.ghiotto, sebastiano.giacomini 2023. Open Science for Social Sciences and Humanities: Open Access availability and distribution across disciplines and Countries in OpenCitations Meta - PROTOCOL. **protocols.io** https://dx.doi.org/10.17504/protocols.io.5jyl8jo1rg2w/v5Version created by maddalena.ghiotto

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

Created: Aug 13, 2023

Last Modified: Aug 16, 2023

PROTOCOL integer ID: 86423

Keywords: ERIH PLUS, SSH journals, OpenCitations Meta, Social Sciences, Humanities, Disciplines coverage, disciplines

Open Science for Social Sciences and Humanities: Open Access availability and distribution across disciplines and Countries in OpenCitations Meta - PROTOCOL V.5

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

[1]University of Bologna

Ali Ghasempouri: ORCID: 0000-0003-3446-1115
maddalena.ghiotto: ORCID: 0009-0009-1309-6340
sebastiano.giacomini: ORCID: 0009-0007-7813-0939

maddalena.ghiotto

ABSTRACT

In this study, we present the workflow followed for the research workflow to assess the coverage of publications in Social Sciences 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 various datasets 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.

Abstract of the research:
Purpose: This study aims to investigate the representation and distribution of Social Science and Humanities (SSH) journals within the OpenCitations Meta database, with a particular emphasis on their Open Access (OA) status, as well as their spread across different disciplines and countries. The underlying premise is that open infrastructures play a pivotal role in promoting transparency, reproducibility, and trust in scientific research.
Study Design and Methodology: The study is grounded on the premise that open infrastructures are crucial for ensuring transparency, reproducibility, and fostering trust in scientific research. The research methodology involved the use of secondary data sources, namely the OpenCitations Meta database, the ERIH PLUS bibliographic index, and the DOAJ index. A custom research software was developed in Python to facilitate the processing and analysis of the data.
Findings: he results reveal that 78.1% of SSH journals listed in the European Reference Index for the Humanities (ERIH-PLUS) are included in the OpenCitations Meta database. The discipline of Psychology has the highest number of publications. The United States and the United Kingdom are the leading contributors in terms of the number of publications. However, the study also uncovers that only 38% of the SSH journals in the OpenCitations Meta database are OA.
Originality: This research adds to the existing body of knowledge by providing insights into the representation of SSH in open bibliographic databases and the role of open access in this domain. The study highlights the necessity for advocating OA practices within SSH and the significance of open data for bibliometric studies. It further encourages additional research into the impact of OA on various facets of citation patterns and the factors leading to disparity across disciplinary representation.

Data Used

1

- **ERIH PLUS** index of Social Sciences and Humanities approved journals dataset (2MB, downloaded 27/04/2023)

Dataset

ERIH NAME

https://kanalregister.hkdir.no/publiseringskanaler/erihplus/periodical/listApproved LINK

- **OpenCitations Meta** data dump (36 MB, downloaded in the version of 24/02/2023)

Dataset

OC Meta NAME

https://opencitations.net/download#meta LINK

- **DOAJ**, the Directory of Open Access Journals public dump (22 MB, downloaded 28/05/2023)

Dataset

DOAJ NAME

https://doaj.org/docs/public-data-dump/ LINK

1.1 **ERIH**

This dataset has the following columns:

- Journal ID: the unique ERIH PLUS assigned Journal identifier
- Print ISSN
- Online ISSN
- Original Title
- International Title
- Country of Publication
- ERIH PLUS Disciplines
- OECD Classifications
- [Last Updated]

	Journal ID	Print ISSN	Online ISSN	Original Title	International Title	Country of Publication	ERIH PLUS Disciplines	OECD Classifications	[Last Updated]
	486254	1989-3477	NaN	@tic.revista d'innovació educativa	@tic.revista d'innovació educativa	Spain	Interdisciplinary research in the Social Sciences, Pedagogical & Educational Research	Educational Sciences; Other Social Sciences	2015-06-25 13:48:26

OC META

- id
- title
- author

- issue
- volume
- venue
- page
- pub_date
- type
- publisher
- editor

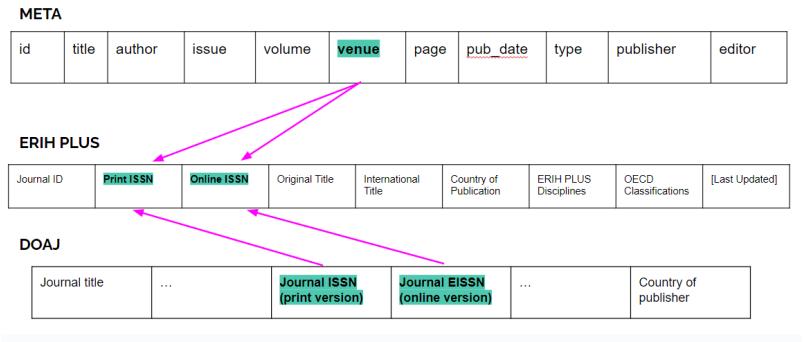
id	title	author	issue	volume	venue	page	pub_date	type	publisher	editor
meta:br/060209 doi:10.4230/lipics.approx/random.2020.19	Distributed Testing Of Graph Isomorphism In The CONGEST Model	Levi, Reut [meta:ra/0610110096 orcid:0000-0003-3167-1766]; Medina, Moti [meta:ra/0612046435 orcid:0000-0002-5572-3754]			[meta:br/060182 issn:1868-8969]		2020	report	Schloss Dagstuhl - Leibniz-Zentrum Für Informatik [meta:ra/0605251]	Byrka, Jarosław [meta:ra/069044096 orcid:0000-0002-3387-0913]; Raghu Meka [meta:ra/0605252]

DOAJ

This dataset includes 55 columns, mainly with data about different aspects of Open Access. For the sake of our research we only needed few columns:

- Journal ISSN (print version)
- Journal EISSN (online version)
- Country of publisher

1.2 To answer our research questions we have to find a way to connect the three datasets. We do so by merging them on the basis of the common issn values



Process OC META, ERIH and DOAJ

1.3 **APPROACH FOLLOWED TO ANSWER RQ1:** *What is the coverage of publications in Social Sciences and Humanities (SSH) journals (according to ERIH PLUS) included in OpenCitations Meta?*
and RQ4: *How many of the SSH journals are available in Open Access according to the data in DOAJ?*

The class **PlayaristsProcessor** is created to answer RQ1.
It is initialized with the following user modifiable attributes as parameters:

- **batch_size**: the size of the csv batches to process
- **max_workers:** the number of cpu used for processing

- **meta_path**: the path to the folder containing all the OC Meta dump csv files
- **erih_path**: the path to ERIH dataset
- **doaj_path**: the path to DOAJ dataset

the _init_ function:
- loads ERIH csv in a pandas Dataframe
- loads DOAJ csv in a pandas Dataframe
- finds all the files with extension ".csv" in the folder specified by meta_path and creates a list of file paths pointing to each CSV file in the folder.

The class has a main method **process_files** in charge of filtering and merging the three datasets, with the aid of some ancillary functions z(described in section 2.2).
It computes the following steps:

1. it creates a dictionary from ERIH dataset
- iterates over the rows of the datasets
- for every row it creates two key-value pairs, one with the print issn as key and the erih unique identifier as value and the other with the online issn as key and the erih unique identifier as value
- it adds each key-value pair to the dictionary "erih_plus_dict"

2. it processes meta batches in parallel
- creates an empty list "all_results"
- sets a variable to keep track of the total publication count as 0
- loops over a sequence of numbers "i", incrementing by the specified batch files number at every loop
- inside the loop, it generates a "meta_path" sublist called "batch_files" which is the list of csv paths going from index "i" to the index of the last csv in the batch. (ex. if "batch_size" is 150, in the first iteration "batch_files" contains the first 150 csv paths, the second iteration it contains the next 150 and so on)
- uses ProcessPoolExecutor for executing functions asynchronously in a separate process, specifying the maximum number of worker processes
- In the ProcessPoolExecutor maps the function **process_file_wrapper** (section 2.2 of the workflow) to each file in the "batch_files" list concurrently. The function is given as argument a tuple containing a single file of the list and ERIH dictionary previously created.
- **process_file_wrapper** returns a tuple with the file name, the processed batch as a dataframe, the number of publications counted for each journal processed.
- at every iteration of the loop this resulting tuple is appended to to the list "all_results".

```
for i in range(0, len(self.meta_path), self.batch_size):
    batch_files = self.meta_path[i:i+self.batch_size]
    with concurrent.futures.ProcessPoolExecutor(max_workers=self.max_workers) as executor:
        results = executor.map(process_file_wrapper, [(f, erih_dict) for f in batch_files])
        all_results.extend(results)
```

3. Generates a preliminary processed dataframe (in this protocol, we will refer to this as **"meta_erih_processed"**) containing the OC Meta unique identifier, the issn(s), the ERIH unique identifier, and the count of publications in each venue:

	OC_omid	issn	EP_id	Publications_in_venue
	06101064393	"['2619-1008', '2619-0990']"	494990	26

- iterates over "all_results" list and disassemble the tuple
- creates a dictionary with the file names as keys and the corresponding dataframe as value
- adds the publication counted for that batch to the total publication count
- all the values in the dictionary, namely, all the dataframes are concatenated to create the final dataframe.
- groups the rows of the dataframe based on the unique combinations of values in the columns corresponding to OpenCitations omid, the issn and the ERIH identifier, creating groups where rows with the same values in these columns are placed together. The corresponding publication counts are summed.

4. Retrieves Open Access information by processing DOAJ dataset (using a function **process_doaj_file**)
- adds a column named "Open Access" that has "unknown" as value if the journal is not found in DOAJ, otherwise has value "True"

	OC_omid	issn	EP_id	Publications_in_venue	Open_Access
	06101064393	"['2619-1008', '2619-0990']"	494990	26	"True"

5. Cleans double entries of the same Journal
We found that in OC Meta are present instances of the same Journal with different omid identifiers, where ERIH only presents one (ex. a print issn and an online issn of a journal are connected to two separate omids). For this reason, we need to clean our result dataset from double entries and sum their publication count.

- the duplicted "EP_id" values are detected with **duplicated** function which creates a list of booleans with a True value for every row that is a duplicated (we detect both of the duplicates).
- the list is temporarily added as a column to the dataframe and a subdataframe with only duplicates is created by queerying for True values.
- A new subdataframe is created from the duplicated one, to keep only the columns with OC omid and the issn related
- this dataframe is saved as a research result dataset in csv format, as it might be useful for further inquiry into this peculiarity.

Dataset

duplicate_omids NAME

https://doi.org/10.5281/zenodo.8249907 LINK

- by aggregating the main dataframe on EP_id and summing corresponding publications count, a temporary dataframe is created with these two values
- this dataframe is merged again with the main one on the common EP_ids keeping only the entries in the aggregated one, hence discarding the doubles.
- finally, the dataframe is sorted by publications' count in descending order and saved as a result dataset in csv format.

Dataset

SSH_Publications_in_OC_Meta_and_Open_Access_status NAME

https://doi.org/10.5281/zenodo.8249907 LINK

ANSWER RQ1 (W*hat is the coverage of publications in Social Sciences and Humanities (SSH) journals (according to ERIH PLUS) included in OpenCitations Meta?*)**:**
To answer the research question the percentage of the publications in ERIH covered by OC_Meta is calculated by summing all the numbers of publications for each venue in SSH_Publications_in_OC_Meta_and_Open_Access_status and dividing it by the total publication count returned by **process_file_wrapper.** The division is multiplied by 100 to compute the percentage.

ANSWER RQ4(*How many of the SSH journals are available in Open Access according to the data in DOAJ?*)**:**
We count the rows that have a "True" value in the "Open Access" column of SSH_Publications_in_OC_Meta_and_Open_Access_status.csv. We compute the percentage by multiplying this count per 100 and dividing by the total number of rows in the dataset

1.4 ANCILLARY FUNCTIONS TO PROCESS THE FILE

1. **process_file_wrapper**

- takes the input file (a single csv file of the batch) and the ERIH dictionary, disassemble them from the tuple they were in and passes them to the **process_file** function.

2. **process_file** processes a large csv file in chunks

- sets the chunksize to 5000 rows
- initializes an empty list to store the processed files and a counter to keep track of the total number of processed publications per chunk
- initialized an empty dataframe to store the combined results
- it reads the input csv in chunks and processes each chunk separately by iterating over all the chunks
- in the loop the chunk, along with the ERIH dictionary, is passed to the function **process_meta_csv** that returns a tuple containing a dataframe with the processed chunk and the number of processed publications in the chunk
- the processed chunk is appended to the list and the counter is incremented by the number of publications processed
- the processed chunk dataframe is concatenated to the empty dataframe initialized before

- after the iterations are finished, the function returns a tuple with the final dataframe and the number of processed rows to **process_file_wrapper** that returns the processed file name, the processed batch as a dataframe, the number of publications counted for each journal processed to the main **process_files**

3. **process_meta_csv**

RETRIEVING VENUES' ISSNS AND COUNTING THE NUMBER OF PUBLICATION PER VENUE

- counts the number of rows in the chunk where the "id" column contains the string 'doi', indicating that these rows are publications. then stores them in a variable
- then, converts the "venue" column to string data type and extracts issn numbers with **findall** method, using the regular expression:

```
"issn:(\d{4}-\d{3}[\dXx])"
```

- adds to the csv a column "issn_list" with the list of retreived issn and an empty list if nothing is found
- extracts the omid identifier from the "venue" column using regular expression and stores them in a new "OC_omid" column

```
":br/([^\s]*)"
```

- drops duplicates from the OC Meta dataframe based on "issn_list" column values to obtain a dataframe with unique issn lists
- takes the indexes of the rows of the unique issn lists, and creates a series of "OC_omid" values corresponding to those indexes, so that all omids related to the issns are retrieved. For practicality, we'll refer to it as "OMID_series"
- substitutes the column "issn" with the column "issn_list"
- aggregates the values in the dataframe by issn, and the values in the "id" column (the publications) are counted so that the number of publications for venues are retrieved and assigned to the "id" column.
- the values of "OMID_series" are then assigned back to the "OC_omid" column
- the "id" column is renamed "Publications_in_venue"

MERGING WITH ERIH DATA

- the function initializes an "EP_id" column with empty values
- and an empty set to store the indexes of the OC Meta issn found in ERIH
- then loops over the values of the "issn" column in the processed OC Meta. Since the value is a string that looks like this,

```
"['2215-1486', '2215-1478']"
```

to access these values as list elements, the code removes the brackets and splits the string at the comma, then, replaces the apostrophes with an empty string.

- for each ISSN, it checks if it exists in the ERIH dictionary keys. If it does, the corresponding "EP_id" is assigned to the "EP_id" column and the dataframe index of that OC Meta issn is added to the set initialized before
- a new dataset is created by selecting only the rows in the processed OC Meta that correspond to the indexes of the OC Meta issn found in ERIH
- finally, all unnecessary columns that were still there from the original OC Meta are excluded from the dataframe and only the following are kept: 'OC_omid', 'issn', 'EP_id', 'Publications_in_venue'
- the final dataframe and the counted publications are returned to **process_file**

1. **process_doaj_file**

This function is called to add Open Access status information to the **meta_erih_processed** dataframe (section 1.3).

- it first filters the DOAJ dataframe by selecting three columns: the 5th, 6th, and 10th columns. The columns correspond to print issn, online issn, and Country of publisher
- it then creates an empty dictionary, which will map both print and online issns in DOAJ 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
- outside the loop, It creates a list of all the keys (ithe issn of venues that are Open Access) in the dictionary of DOAJ issn previously created
- it then adds a new column, **'Open Access'**, to the **meta_erih_processed** dataframe and initializes all its values to "Unknown".
- the function then iterates over each ISSN in the **meta_erih_processed** dataframe. If this identifier is found in the list of Open Access identifiers, the Open Access status of that row in **meta_erih_processed** is updated to True.
- finally, **meta_erih_processed** with added Open Access information is returned to **process_files**

Retrieving Data about Countries and Disciplines

2 **APPROACH FOLLOWED TO ANSWER RQ2 and RQ3:** *What are the disciplines that have more publications? What are countries providing the largest number of publications and journals?*

A similar approach is followed to find the disciplines with more publications and the countries with more publications and journals.

- The primary souce for these information is ERIH dataset, the interested columns are "ERIH PLUS Disciplines" and "Country of Publication".

1. RETREIVING WHICH JOURNALS BELONG TO THE SAME PER DISCIPLINE

This is done by creating a dictionary that maps each discipline to a list of ERIH journal IDs associated with that discipline. The class created for this part is **DisciplinesProcessor** with the inner method **create_disciplines_dict** that does the following:

- an empty dictionary is initialized
- ERIH dataset is merged with SSH_Publications_in_OC_Meta_and_Open_Access_status on the basis of the common ERIH identifier
- a loop iterates over this merged dataframe:
- it checks whether the dictionary is empty or not
- if it's empty it takes the values in "ERIH PLUS Disciplines" column (a string with possibly many disciplines separated by a comma) and creates a list with all the disciplines as items, using the **split(', ')** method. Then, it adds every discipline as dictionary key and the ERIH identifier of the considered row as value
- if the dictionary is not empty, it takes the values in "ERIH PLUS Disciplines" column and creates a list with all the disciplines as items, then it transforms the list in a set
- it initializes a set with the keys (the disciplines) already present in the dictionary and computes the difference between the rows' disciplines and the disciplines in the set created from the dictionary. In this way, it saves in a set "diff" all the new disciplines to add to the dictionary
- then it checks if "diff" is populated with something or not
- if the set is empty it means that all the disciplines are already keys of the dictionary, so it simply appends the row's ERIH identifier to the list values of the interested disciplines
- otherwise it updates the dictionary with the new disciplines in the "diff" sets and add the row's ERIH identifier to the list value of all the keys corresponding to disciplines found in "ERIH PLUS Disciplines".
- when the iterations are finished, the complete disciplines' dictionary is returned

The file is saved in json format

Dataset

result_disciplines NAME

https://doi.org/10.5281/zenodo.8249907 LINK

2. 1. RETREIVING WHICH JOURNALS BELONG TO THE SAME PER COUNTRY

This is done by creating a dictionary that maps each country to a list of ERIH journal IDs associated with that country. The class created for this part is **CountriesProcessor** with the inner method **create_counties_dict** that does the following:

- an empty dictionary is initialized
- an empty list "unmatched_countries" is initialised. This list will be filled with the identifiers of the journals for which, at the end of the process, no country has been retrieved
- ERIH dataset is merged with SSH_Publications_in_OC_Meta_and_Open_Access_status on the basis of the common ERIH identifier
- a loop iterates over this merged dataframe:
- as a first thing, the function checks is the value of the row "Country of Publication" in the dataframe is empty
- if it's empty, no country can be retrieved and the ERIH identifier is added to the list "unmatched_countries"
- otherwise, it checks whether the dictionary is empty or not
- if it's empty it takes the values in "Country of Publication" column (a string with possibly more than one countries separated by a comma) and creates a list with all the countries as items, using the **split(', ')** method. Then, it adds every country as dictionary key and the ERIH identifier of the considered row as value
- if the dictionary is not empty, it takes the values in "Country of Publication" column and creates a list with all the countries as items, then it transforms the list in a set
- it initializes a set with the keys (the countries) already present in the dictionary and computes the difference between the rows' countries and the countries in the set created from the dictionary. In this way, it saves in a set "diff" all the new countries to add to the dictionary
- then it checks if "diff" is populated with something or not
- if the set is empty it means that all the countries are already keys of the dictionary, so it simply appends the row's ERIH identifier to the list of the interested countries
- otherwise it updates the dictionary with the new countries in the "diff" sets and add the row's ERIH identifier to the list value of all the keys corresponding to countries found in "Country of Publication".

After iterating over the whole dataset it is possible that there are journals with no country specified, these are stored in the empty list

"unmatched_countries". To try and retrieve a country for these journals we look in the DOAJ dataset, where there is also a column "Country of publisher". The function called to fetch the missing countries from DOAJ is **retrieve_doaj_country** and it works as follows:

- it filters the merged dataframe (ERIH + SSH_Publications_in_OC_Meta_and_Open_Access_status) to contain only rows where the EP_id is also in the "unmatched_countries" list.
- it uses the method **explode** on the dataframe to transform each element of a list-like (the issn's list) to a row, replicating all the other column values of that row, for as many issn where present in the list, so that every row generate looks like the initial one, with the exeption of the "issn" column, that has a single different issn as value
- then the dataframe is merged twice with DOAJ, one time on the common print issn, and one on the common online issn. In both cases this is done using only keys from the merged dataframe and not from DOAJ. The generated dataframe has two columns "Country_of_publisher_x" and one "Country_of_publisher_y", one coming from the print issn and one for the online issn
- a column "Country" is created to unify these country values, this is done by filling the empty cells in "Country_of_publisher_x" with the value, if found, in "Country_of_publisher_y" and assigning the values to the new "Country" coulmn
- all the nan values in "Country" column are then dropped to the dataframes. These are the ERIH identifiers that are still left without venue. These identifiers are also stored in a list that will not be saved but will be printed out when running the software.
- Now the created countries dictionary is extended with the new information. This is done by iterating over rows of the dataframe and checking if the "Country" value is in the country dictionary
- if it's not, the new key-value pair is created
- otherwise, the new identifier is appended to the list of that country's value

In this step it's necessary to troubleshoot instances where the same country is referred to in different ways. There are two of these occurrences:
'Turkey' and 'Türkiye' ; 'Venezuela' and 'Venezuela, Bolivarian Republic of'.

- the code checks if both nomenclatures are dictionary keys and extend the list of identifiers of the first with those of the second. Then deletes the key with the second nomenclature

Finally the instance of a key "Republic of" erroneously created by using the **split** method on strings like "Korea, Republic of" is deleted.
The completed country dictionary and the list of unmatched identifiers are returned as a tuple. The countries' dictionary is saved as a file in json format.

Dataset

NAME	LINK
result_countries	https://doi.org/10.5281/zenodo.8249907

2.1 COMPUTING THE RESULTS

Finally, to create the datasets that answer our research question a class **CountsProcessor** with the inner method **counts** that does the following:

- takes in input a dictionary and a label: the dictionary is either **result_countries** or **result_disciplines,** the label is a string that is going to define the name of the column where the values of disciplines or countries will be stored
- creates a new empty dataframe with three columns, one identified by the input label, "Journal_count" and "Publication_count" to store the counts, respectively of journals and publications for each vountry or each discipline. For practicality we will refer to it here as **count_df**
- iterates iver the input dictionary items, for each key-value pair, meaning for each unique discipline or country (the keys of the dictionaries), it performs the following operations
- filters SSH_Publications_in_OC_Meta_and_Open_Access_status to contain only the issn in the list which is the dictionary's value
- counts the number of journals retreived
- counts the number of publication by summing the the publication count for each journal = the values of the column "Publications_in_venue"
- creates a dataframe row of **count_df** setting the key of the dictionary as value of the user labeled column, the count of journals as value of "Journal_count" and the count of publications as value of "Publication_count".
- at each iteration through the dictionary items, the new row is concatenated to the **count_df**
- once iterations are over, the dataframe is sorted in descending order and saved in csv format.

Two datasets are created by running this code first with the countries dictionary and then with the disciplines dictionary:

Dataset

SSH_Publications_and_Journals_by_Country NAME

https://doi.org/10.5281/zenodo.8249907 LINK

Dataset

SSH_Publications_by_Discipline NAME

https://doi.org/10.5281/zenodo.8249907 LINK

2.2 COMPUTING WITHOUT MEGA JOURNALS

In the second version of the software, we realized that the presence of Mega Journals might be skewing our results, since a significant disparity is evident between the publication count of the top four largest journals Science, Nature, PNAS, PLOS and all other journals. These Journals are known to cover a broad range of subjects beside SSH subjects and publish extensively more than other journals. Since in our methodology all of the publications of a journal count as belonging to each of the journal's discipline, the disciplines covered by mega journal might erroneously increase their rank.

- In the second version of the software, we incorporated a feature that enables the exclusion of these four mega journals from the analysis. This enhancement facilitates more comprehensive comparisons. To achieve this, a new parameter named "remove_megajournals" has been added, to the arguments of of the following classes: **DisciplinesProcessor, CountriesProcessor,** and **CountsProcessor**. The parameter takes boolean values that can be specified by the user while launching the program. If set as True, when starting to process disciplines and countries, it will remove the first four rows of SSH_Publications_in_OC_Meta_and_Open_Access_status

Comparing EU values with US values

3 In the second version of the software, a more detailed comparison of US Journals and European Journals has been carried out, to better understand how disciplines are distributed in the two countries with the highest number of publications.

The class **Compare_US_EU**, was created to perform such task, with two internal methods: **compare_us_eu** and **counts_us_eu**.

The first method, **compare_us_eu**, takes in input the ERIH dataset and the Countries dictionary, filters the dictionary to retrieve the values (the ERIH identifiers) for European countries and "United States" and stores them in two separate lists called "eu_ID" and "us_ID".
As a criterion, cases where two European countries are specified as the country of publication of the same journal are kept, while Turkey and Russia are excluded.
Then, it filters the ERIH dataset, and creates two new temporary Dataframes, one only with US Journals and one with European Journals.
To do so, the .isin() method is used to check whether an ERIH identifier in the dataset is present in the two lists previously created of US and European identifiers.
Subsequently, for both the created Dataframes, the number of disciplines per Journal mentioned in the column "ERIH PLUS Disciplines" are counted and the count is added as a new column "disc_count" value.
After this step the two Dataframes are merged with SSH_Publications_in_OC_Meta_and_Open_Access_status through the common ERIH identifier to add data about the number of Publications in venue; the intersection between the two is obtained by using intersection of keys from both frames and the result Dataframes are named "us_meta" and "eu_meta".
On these two Dataframes the following operations are performed:

- they are filtered to create two datasets, one for US and one for Europe, containing only the columns "EP_id", "Publications_in_venue", "Original_Title", "Country_of_Publication", "ERIH_PLUS_Disciplines", "disc_count"
- exported as csv file with the name of "**us_data**" and "**eu_data**", useful to record meaningful information that are excluded from the other datasets

Dataset

us_data NAME

https://doi.org/10.5281/zenodo.8249907 LINK

Dataset

eu_data NAME

https://doi.org/10.5281/zenodo.8249907 LINK

- They are filtered to create two other datasets, again, one for US and one for Europe, containing only the columns that were in SSH_Publications_in_OC_Meta_and_Open_Access_status
- they are exported in csv with the name "**meta_coverage_us**" and "**meta_coverage_eu**", necessary to have a data structure fit for the computation of the disciplines count described below.

Dataset

meta_coverage_us NAME

https://doi.org/10.5281/zenodo.8249907 LINK

Dataset

meta_coverage_eu NAME

https://doi.org/10.5281/zenodo.8249907 LINK

The second method, **counts_us_eu**, works exactly as the **counts** method described in section 2.1, but takes either "meta_coverage_us" or "meta_coverage_eu" as parameter too and performs the count on them. The results are also exported as 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.
- a barchart comparing the publications per discipline of European Countries and the United States