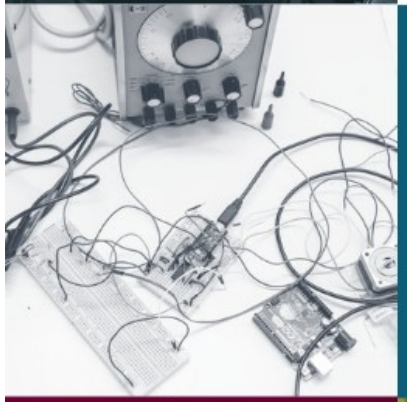
Sherman Centre for Digital Scholarship



Assessing the Value of Transformative Agreements

An Automated Approach to Estimating Article Processing Charge (APC) Discounts

Jack Young, Bibliometrics & Research Impact Librarian

Kelvin Lee, Research Software Developer

June 5, 2025

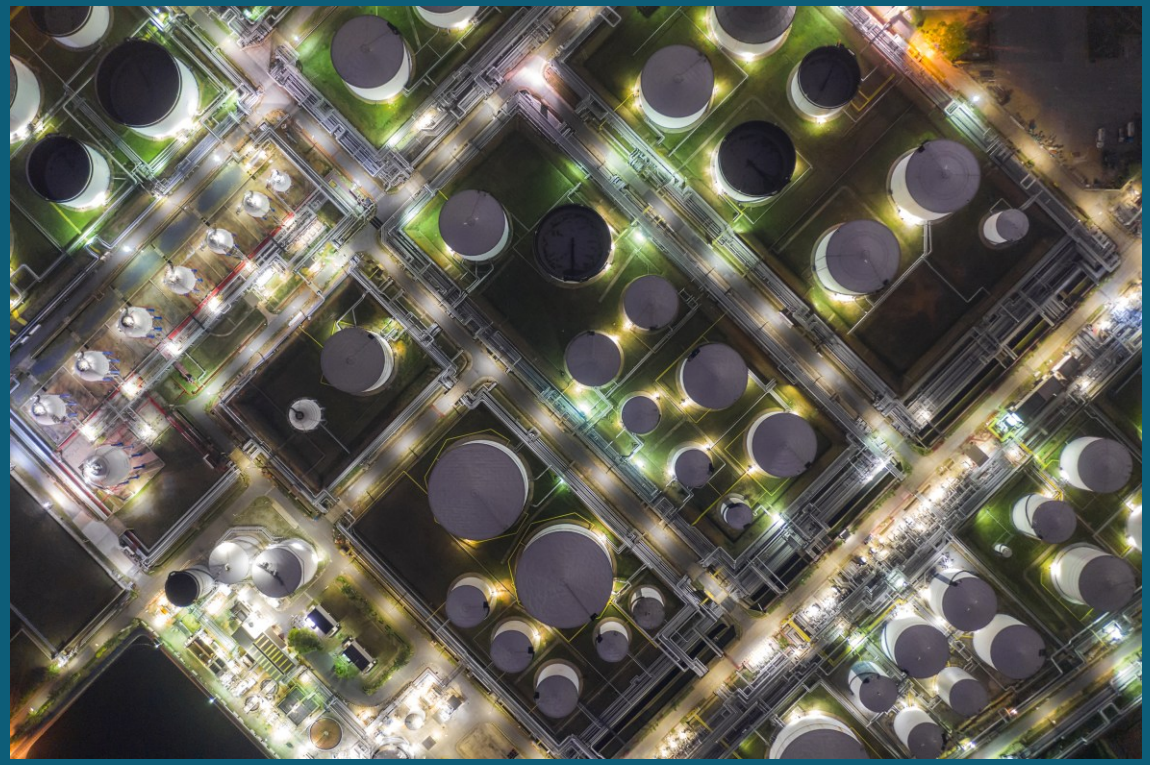
McMaster University Library

Land Acknowledgement

McMaster University is located on the traditional Territories of the Mississauga and Haudenosaunee Nations, and within the lands protected by the “Dish With One Spoon” wampum agreement.

Session Overview

- Context
- Methodology
- Results
- Challenges/Limitations
- Next Steps
- Questions/Feedback



Context

AskResearch at McMaster

AskResearch is a library-hosted service that brings digital research support teams at McMaster together under a single service point. Component teams include:

- Research Data Management
- Information Security
- High-Performance Computing
- Research Software Development and Support
- Research Impact

Helps simplify navigation for researchers and fosters collaboration between teams (i.e. Software Development + Research Impact)

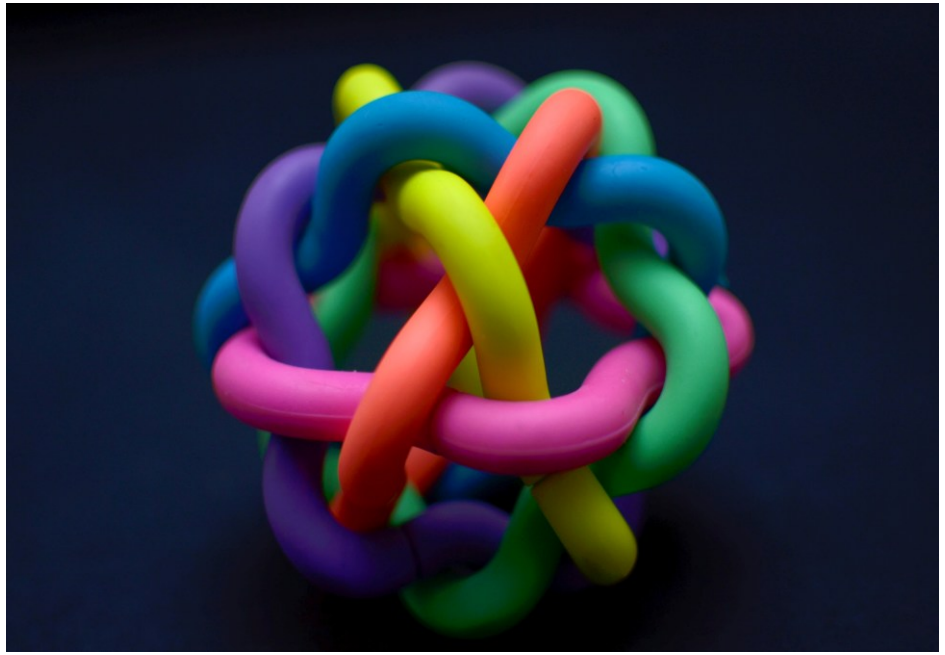
Transformative Agreements

Transformative (read-publish) agreements offer waived or discounted article processing charges (APCs) when authors from the participating organization publish open access.

In 2024, McMaster was part of the Canadian Research Knowledge Network's (CRKN) consortia-led transformative agreement with 20 publishers.

Being able to confidently assess and express the value of such agreements is important for today's research libraries, especially as Universities tighten their budgets.

Although some publishers provide transparent APC discount data to libraries, there is still significant value in libraries keeping their own records of this data.



scds.ca
scds@mcmaster.ca

Methodology

OpenAlex

OpenAlex is an open bibliographic database of over 240m scholarly outputs.

- Free to use (both UI and API).
- Well integrated with other open data sources (e.g. DOAJ, CrossRef, RoR, etc.)
- “Corresponding Institution” and “APC” fields made it uniquely suited to this project.

Computational Notebooks

What they are, why we used them?

- An interactive, document-based environment
- Write and execute (data querying and analysis) code
- Document steps and rationales
- Visualize results
- Notable examples include Jupyter Notebook and Google Colab

Parameters

- `corresponding_institution_ids` (McMaster's OpenAlex ID - dervied from ROR)
- `publication_year` (2024)
- `type` (`article` or `review`)
- `oa_status` (`gold` or `hybrid`)

Code Snippet

```
ror_id = "https://ror.org/02fa3aq29"
publication_year = 2024
publication_types = ["article",
"review"]
publication_oa_statuses = ["gold",
"hybrid"]
```

Sherman Centre for Digital Scholarship | Library | McMaster University

Query Publications from OpenAlex

```
def get_works_by_corresponding_institution(institution_id, publication_year, publication_types, page=1, items_per_page=50):
    url = f"""
        https://api.openalex.org/works?filter=
        corresponding_institution_ids:{institution_id},
        publication_year:{publication_year},
        type:{'|'.join(publication_types)},
        oa_status:{'|'.join(publication_oa_statuses)}&
        page={page}&per-page={items_per_page}
    """

    response = requests.get(url)
    json_data = response.json()

    df_works = pd.DataFrame.from_dict(json_data["results"])

    next_page = True
    if df_works.empty:
        next_page = False

    if next_page:
        df_works_next_page = get_works_by_institution(
            institution_id,
            publication_year,
            publication_types,
            page=page+1,
            items_per_page=items_per_page
        )
        df_works = pd.concat([df_works, df_works_next_page])

    return df_works

df_works = get_works_by_corresponding_institution(institution_id, publication_year, publication_types, publication_oa_statuses)
```

Apply Necessary Transformation and Clean Up

```
# extract 'value_usd' from 'apc_list' if it is a dictionary (i.e. 'apc_list' exists in the work record); otherwise, set to null
df_works["apc_usd"] = df_works["apc_list"].apply(lambda apc_list: apc_list["value_usd"] if isinstance(apc_list, dict) else np.nan)

# extract 'id' and 'name' from 'source' within 'primary_location' if 'source' exists; otherwise, set to null
df_works["source_id"] = df_works["primary_location"].apply(lambda location: location["source"]["id"] if location["source"] else np.nan)
df_works["source_name"] = df_works["primary_location"].apply(lambda location: location["source"]["display_name"] if location["source"] else
np.nan)

# extract 'host_organization' from 'source' within 'primary_location' if 'source' exists; otherwise, set to null
df_works["source_host_organization"] = df_works["primary_location"].apply(lambda location: location["source"]["host_organization"] if
location["source"] else np.nan)

# extract 'issn' and 'issn_l' from 'source' within 'primary_location' if 'source' exists; otherwise, set to null
df_works["source_issn"] = df_works["primary_location"].apply(lambda location: location["source"]["issn"] if location["source"] else np.nan)
df_works["source_issn_l"] = df_works["primary_location"].apply(lambda location: location["source"]["issn_l"] if location["source"] else
np.nan)

# calculate the average apc where 'apc_list_usd' is not null
apc_mean = df_works[df_works["apc_list_usd"].notnull()]["apc_list_usd"].mean()

# fill null values in 'apc_list_usd' with the calculated average
df_works["apc_list_usd"] = df_works["apc_list_usd"].fillna(apc_mean)

# fill null values in 'source_id', 'source_name', 'source_issn' and 'source_issn_l'
df_works["source_id"] = df_works["source_id"].fillna("unknown source")
df_works["source_name"] = df_works["source_name"].fillna("unknown source")
df_works["source_host_organization"] = df_works["source_host_organization"].fillna("unknown source")
df_works["source_issn"] = df_works["source_issn"].fillna("unknown source")
df_works["source_issn_l"] = df_works["source_issn_l"].fillna("unknown source")
```

Apply Known Discount to OpenAlex Data

```
def get_discount(issn: typing.List[str] | str, apc: float) -> float:
    if isinstance(issn, str):
        issn = [issn]

    discount_rows = df_discount[df_discount["issn"].isin(issn)]

    if discount_rows.empty:
        return apc

    discount_row = discount_rows.iloc[0]

    if discount_row["is_flatrate"]:
        return apc - discount_row["discount"]
    else:
        return apc * (1 - discount_row["discount"])


df_works["discounted_apc_usd"] = df_works.apply(lambda x:
get_discount(issn=x["source_issn"], apc=x["apc_usd"]), axis=1)
```

Results

scds.ca
scds@mcmaster.ca

2024 APC Estimates (McMaster University)

Total APC Cost (Before Discount):

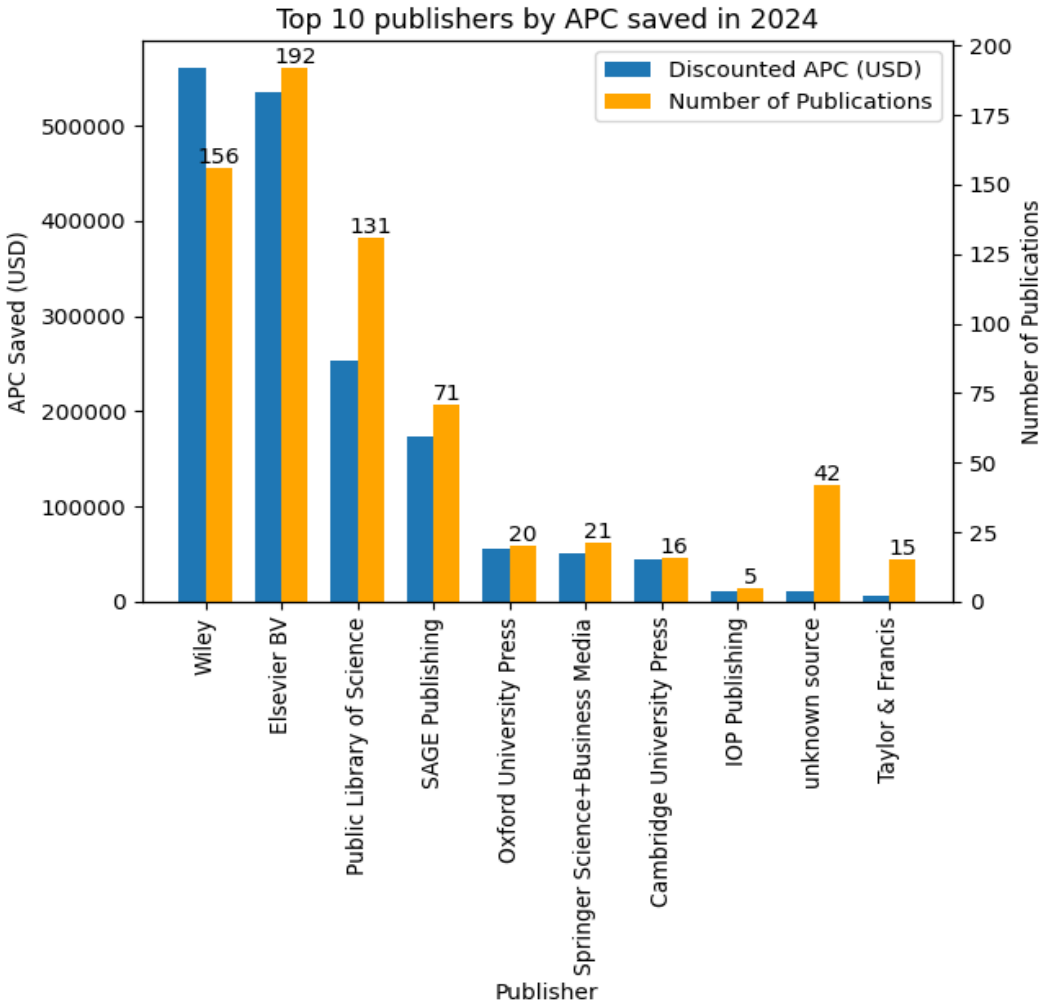
USD $2,741,692.01

Total APC Cost (After Discount):

USD $1,020,350.09

Total APC Savings:

USD $1,721,341.92



scds.ca
scds@mcmaster.ca

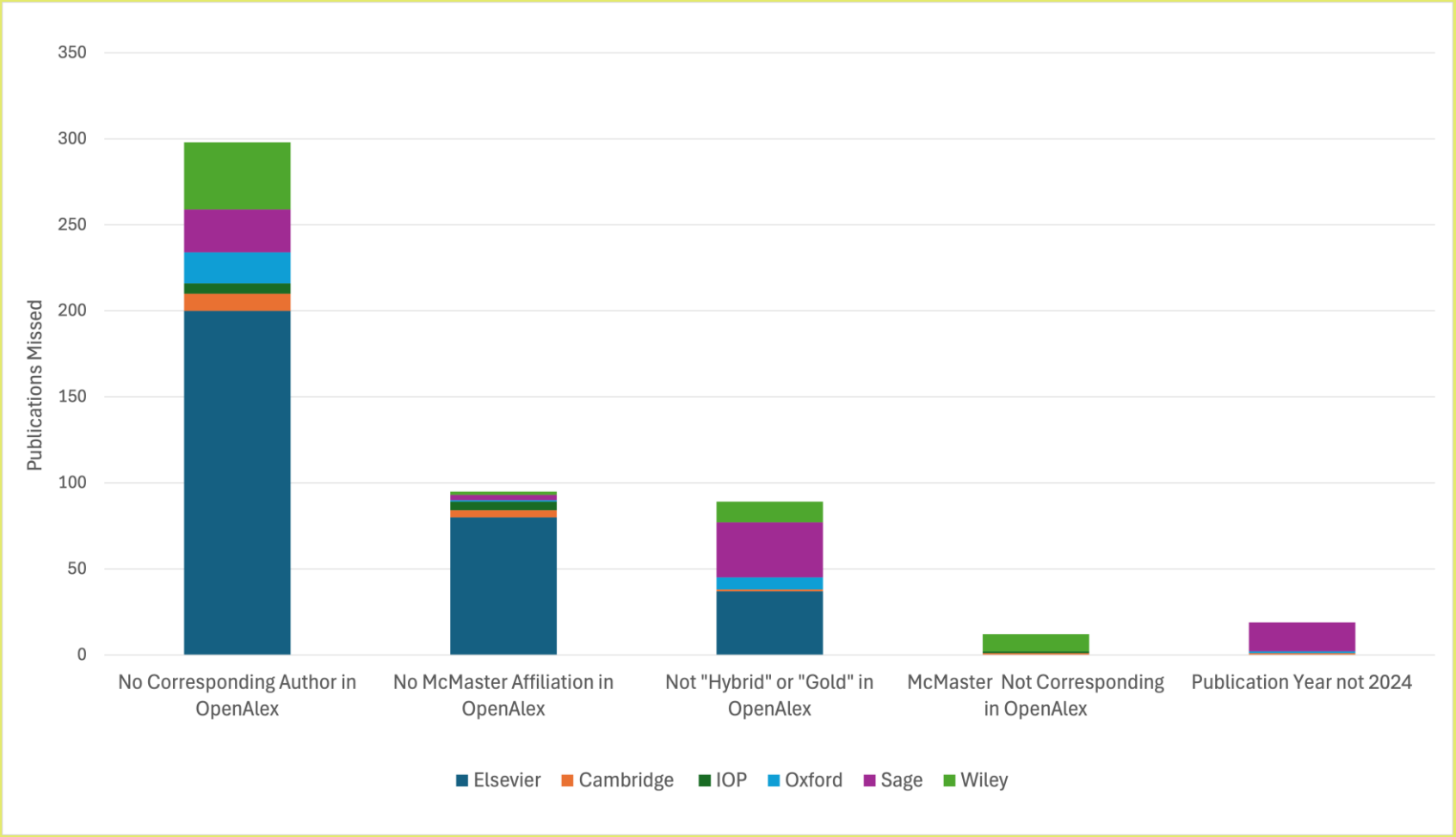
Library | McMaster University | Sherman Centre for Digital Scholarship

Challenges / Limitations

2024 APC Estimates vs. Publisher-Reported

Publisher	# of McMaster OA Articles (Known v. Estimate)	APC Saved in USD (Known v. Estimate)
Cambridge University Press	6	$21,395.00
Elsevier - Hybrid	107	$380,961.00
Elsevier - Gold	-7	-$1,194.00
IOP Publishing - Gold	5	$12,615.00
IOP Publishing - Hybrid	3	$9,500.00
Oxford University Press - Hybrid	22	-$51,519.00
Oxford University Press - Gold	-10	-$3,128.00
SAGE Publications - Hybrid	8	$83,550.00
SAGE Publications - Gold	7	$9,622.00
Wiley	39	$195,523.00
Combined Difference	**180**	**$657,325.00**
Combined Difference (as %)	**28.48%**	**30.96%**

Why Were Publications Missed?



scds.ca
scds@mcmaster.ca

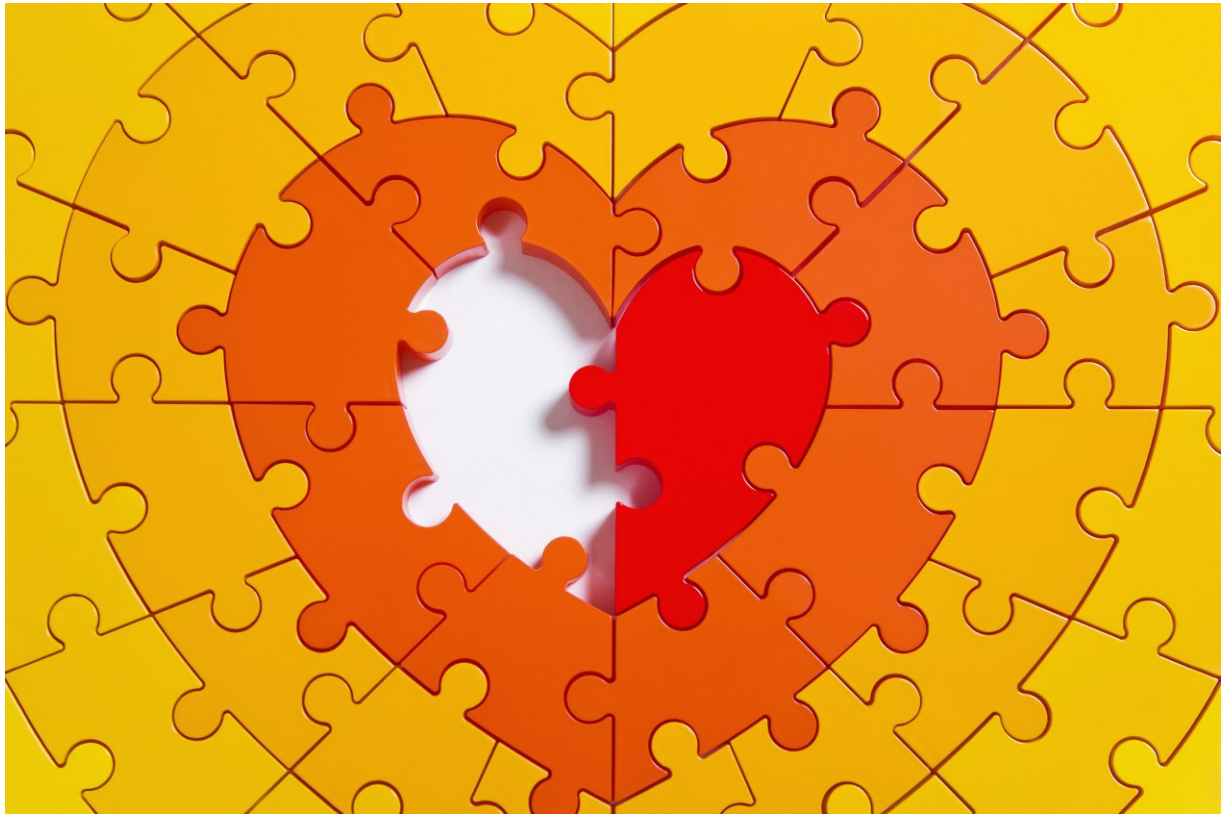
Library | McMaster University | Sherman Centre for Digital Scholarship

APC Data

17% of the retrieved records lacked APC data on DOAJ (mean value of all known APCs was used in these cases).

List APC prices found on publisher websites may not represent the actual amount paid (e.g. promotional rates).

DOAJ APC data is only available for the most recent year, meaning we are unable to apply this approach retroactively.



Next Steps

scds.ca
scds@mcmaster.ca

Next Steps

Meeting with Eric Schares, Leigh-Ann Butler, and Stefanie Haustein to discuss insights from their projects[1,2] tracking APC costs at six major publishers over time. They had some great ideas to workaround some of the challenges we faced:

- When McMaster is the only institution on a publication, consider them corresponding.
- If no corresponding author is listed, consider first author corresponding.
- For unknown APCs, calculate median APC for each publisher / OA type (as opposed to a global mean APC).

[1] Butler, L., Hare, M., Schönfelder, N., Schares, E., Alperin, J. P., & Haustein, S. (2024). An open dataset of article processing charges from six large scholarly publishers (2019-2023). arXiv [Cs.DL]. Retrieved from http://arxiv.org/abs/2406.08356

[2] P., Hare, M., Butler, L.-A., & Schönfelder, N. (2024). Estimating global article processing charges paid to six publishers for open access between 2019 and 2023. arXiv [Cs.DL]. Retrieved from http://arxiv.org/abs/2407.16551

Next Steps (cont.)

- Share results of this analysis with OpenAlex to identify, and hopefully improve, some of the issues with corresponding authorship information in the database.

- Our estimates turned up publications *not* included in publishers' lists, suggesting that some authors are not taking advantage of discounts. This could lead to interventions increasing awareness / use of McMaster's discounted rates.

- Integrating these estimates into library collections' data. Exploring the feasibility of a dashboard tracking APC discounts at McMaster.

Sherman Centre for Digital Scholarship

Questions?

Source code repository:
https://github.com/McMasterRS/research-impact-analysis

APC notebook:
https://github.com/McMasterRS/research-impact-analysis/blob/main/apc.ipynb

Contact:

Jack Young – jkyoung@mcmaster.ca

Kelvin Lee – lee887@mcmaster.ca

McMaster University Library