PUBLICATION-SCRAPER REPORT

Parnika Khattri AND Sherry Tee

2024-09-26

Contents

1 PREFACE

As an R user, have you ever wondered how efficiently academic outputs can be tracked and analyzed across various platforms? What if there was a way to automate the retrieval of publications and software outputs, reducing the time and effort researchers and administrators spend gathering this data? These are the questions that drove the development of Publication-Scraper. This R package enables users to collect and analyze research outputs, providing a more efficient approach to academic performance tracking.

This report is written as part of the ETC5543 research project, supervised by Rob J Hyndman and Michael Lydeamore. Throughout the development of the package, we encountered various challenges, particularly in integrating data from different sources like ORCID, Google Scholar, and CRAN. These challenges, however, provided an opportunity to enhance our technical skills and refine our understanding of research automation.

We are grateful for the guidance and support provided by our supervisors, Rob J Hyndman and Michael Lydeamore and our mentor, David Frazier, whose feedback and encouragement were essential to the success of this project. We hope this report offers valuable insights into the creation and application of Publication-Scraper and highlights its potential to streamline research output analysis for academic institutions.

2 Abstract

This project focuses on analyzing and automating the retrieval of academic outputs using Publication-Scraper, an R package developed to streamline research evaluation. Academic contributions, whether in the form of publications or software outputs, play a critical role in shaping the productivity and impact of researchers. This package leverages ORCID IDs, Google Scholar profiles and CRAN (Comprehensive R Archive Network) data to automate the collection of academic outputs for researchers within the Monash EBS department.

The analysis assumes that publication counts and software downloads provide a reliable and straightforward measure of academic output and impact. Two primary areas of investigation were carried out: one focusing on automating the retrieval of these outputs and the other on visualizing the research productivity metrics. Utilising R for data manipulation and web scraping, the package enables the analysis of trends in research productivity by examining publication counts, software downloads and related factors such as publication year and author activity.

The results from the analysis revealed significant patterns in research output, particularly highlighting prolific authors like Athanasios Pantelous and Rob J Hyndman, who lead in publication counts. The package also identified trends in research activity over time, with sharp increases in publication output during certain periods. Additionally, software packages such as fracdiff and tsfeatures were found to be frequently downloaded, indicating a strong focus on time series analysis within the department.

Overall, Publication-Scraper not only automates data collection but also provides valuable insights into the productivity and impact of academic work within Monash EBS.

KEY WORDS: Academic output analysis, ORCID, Google Scholar, CRAN downloads, R package, Web scraping

3 Introduction

In today's data-driven academic landscape, managing and presenting scholarly outputs has become a critical task for both individual researchers and institutions. In this context, ORCID (Open Researcher and Contributor ID) has emerged as a key platform that provides unique identifiers, linking researchers to their academic contributions. Widely adopted across universities and research institutions, ORCID facilitates efficient retrieval of publication data. Alongside ORCID, Google Scholar profiles offer valuable metrics such as citation counts, h-index and publication history, further enabling comprehensive evaluation of academic performance. However, traditional publication tracking must now be complemented by monitoring software outputs, particularly in computational research domains. The Comprehensive R Archive Network (CRAN) serves as a central repository for R packages, making it an essential data source for assessing the impact of software contributions.

Motivated by the growing need to streamline academic performance evaluation, this project introduces Publication-Scraper, an R package developed to automate the retrieval and analysis of research outputs. In large academic departments like Monash EBS, manually collecting and aggregating data across multiple platforms is time-consuming and prone to error. This package addresses these challenges by leveraging ORCID IDs, Google Scholar profiles and CRAN data to automate the process, providing researchers, administrators and analysts with a powerful tool to track publications and software outputs efficiently.

The primary goal of Publication-Scraper is to offer an integrated solution for scraping, cleaning, and visualizing academic output data. The package enables users to retrieve all publications associated with ORCID and Google Scholar profiles, as well as scrape CRAN download statistics to assess software impact. Key metrics, such as the most prolific authors, the most cited works and the top-downloaded software, are visualized to provide insights into academic productivity and influence. Although designed specifically for the Monash EBS department, Publication-Scraper can be adapted for broader use in research institutions worldwide, ensuring scalability and relevance across different academic contexts.

4 Data

The primary dataset used in this project is the manually compiled `orcid_gsid` dataset, which contains information on Monash EBS staff and their respective ORCID and Google Scholar IDs. This dataset was developed by visiting the Monash University staff directory and collecting relevant academic identifiers for each researcher. Where available, the ORCID and Google Scholar IDs were added to the dataset, though some researchers do not have profiles on these platforms, resulting in missing entries.

The dataset is structured as follows, with the following variables included:

- `first_name`: The first name of the individual.
- `last_name`: The last name of the individual.
- `orcid_id`:The ORCID identifier, linking researchers to their publication records.
- `gsuser_id`: The unique identifier for the researcher's Google Scholar profile.

This dataset serves as the foundation for the *Publication-Scraper* package. It allows the package to automate the retrieval of publications and software outputs by mapping researchers to their online academic profiles across ORCID, Google Scholar and CRAN.

The data was manually collected and stored in CSV format. Although some missing values exist due to the unavailability of profiles for certain researchers, the dataset provides a comprehensive tool for tracking the academic contributions of Monash EBS staff. This single dataset integrates easily into the package's functions, ensuring efficient data retrieval and analysis for research evaluation purposes.

5 Methodology

In this project, we manually compiled a dataset of Monash EBS staff, including their ORCID and Google Scholar IDs, by visiting the Monash staff directory. Using this data, we developed the *Publication-Scraper* package to automate the retrieval of research outputs across ORCID, Google Scholar, and CRAN. We utilized the ORCID API, Google Scholar profiles, and CRAN package download statistics to extract academic outputs and associated metrics. The package was designed to support efficient research evaluation and performance tracking.

The following steps were carried out as part of the methodological process:

- **Dataset Compilation**: We manually created the `orcid_gsid` dataset containing the names, ORCID IDs, and Google Scholar IDs for Monash EBS researchers. This dataset serves as the base for linking academic profiles across platforms.
- **ORCID Data Extraction**: Using the `rorcid` package, we retrieved all publications linked to each researcher's ORCID ID, capturing details such as publication titles, dates, and DOIs.
- **Google Scholar Data Retrieval**: We collected publication data from Google Scholar profiles using the `scholar` package. This process included extracting citation counts, h-index, and individual publication metrics for each researcher.
- **CRAN Software Output Analysis**: For researchers with contributions on CRAN, we obtained the download statistics of their R packages using the `pkgsearch` package. This data helped quantify software impact.
- **Combining Data**: We integrated the publication and software data to create a unified dataset of academic outputs, providing a comprehensive view of each researcher's contributions.
- **Exploratory Data Analysis**: Several analyses were conducted to visualize key metrics and explore academic performance patterns:
  - We identified the **top five most prolific authors** in the Monash EBS department.
  - We explored the **publication trends** of these top authors over time, highlighting key periods of productivity.
  - We analyzed the **distribution of publications by year** to observe research activity trends within the department.
  - We visualized the **most frequently used journals** by Monash EBS staff to identify key areas of focus.
  - We examined the **top R packages by download volume** to understand software impact and usage patterns.

Through these steps, the *Publication-Scraper* package offers a detailed view of academic productivity, enabling research administrators and analysts to efficiently track and evaluate research outputs.

6 Package Design and Structure

- Overview of Design The publication-scraper package is designed with a modular and flexible architecture, allowing staff in the EBS department to retrieve and analyze publication data efficiently. It is built using a combination of web scraping techniques, API integration, and data manipulation functions, ensuring that staff in the Monash EBS department can effortlessly extract data from ORCID, Google Scholar, and CRAN.
- Core Functions The core functions of the publication-scraper package are:
- get_publications_from_orcid(): Retrieves all publications associated with a given ORCID ID, providing access to the author's complete list of works.
- get_publications_from_scholar(): Collects publication data from Google Scholar profiles using the Scholar ID, gathering valuable metrics such as citation counts and publication history.
- cran_all_pubs(): Extracts download statistics for CRAN packages, allowing users to evaluate software impact.

-get_all_publications(): Combines the results from ORCID, Google Scholar, and CRAN to provide a consolidated view of a researcher's publications and software outputs.

- Directory Structure

R/: Contains R scripts with functions to retrieve and process publication data. combine.R: Functions to fetch and combine publication data from ORCID and Google Scholar.

separate.R: Functions to fetch data separately from ORCID and Google Scholar. utils.R: Utility functions to search for CRAN package download statistics. data/: Contains datasets.

orcid_gsid.rda: This dataset includes ORCID and Google Scholar IDs manually gathered from Monash EBS researchers.

data-raw/: Scripts for preparing and cleaning raw datasets, used to generate orcid_gsid.rda.

inst/vignettes/: Holds the vignette explaining the package's functionalities and visualizations, offering examples on how to analyze top authors, publication trends, and software downloads.

vignettes/: Contains the same vignette in markdown format for ease of accessibility and understanding.

man/: Contains documentation files for each R script and dataset in the package.

tests/: Contains unit tests to ensure the correct functioning of the package. DESCRIPTION & NAMESPACE: These define package metadata (version, dependencies, etc.) and exported functions.

README.md: Provides a general overview of the package and guides users on installation and basic usage.

Report.Rmd: The detailed report of the project, including goals, objectives, and methodology.

- Dependencies

The publication-scraper package relies on several R packages to deliver its functionality:

Data Manipulation and Cleaning: dplyr, tidyverse Web Scraping: rvest, xml2 API Integration: rorcid, scholar Data Visualization: ggplot2 Data Handling: tibble HTTP Requests: httr

7 Implementation

8 Example Usage

Here's an example demonstrating how to use the publication-scraper package:

9 Documentation and User Guide

10 Result & Conclusion - A clear description of your contributions

The publication-scraper project was structured to ensure a clear and balanced division of work between Parnika and Sherry, with valuable guidance from our hosts Michael and Rob, who provided support in our guidance of work and resolving bugs throughout the project. On the other hand, David played a pivotal role in providing mental support throughout the project, consistently checking in on progress and offering valuable tips on improving the final report writing.

The work began by preparing the dataset, a crucial component of the package. Parnika manually inputted the ORCID IDs of Monash EBS staff, while Sherry handled the task of inputting the corresponding Google Scholar IDs.

Sherry developed the core functions of the package, such as get_publications_from_orcid(), get_publications_from_scholar(), and get_all_publications(), which retrieve and combine publication data from ORCID, Google Scholar, and CRAN. Parnika provided examples that demonstrated how these functions can be applied, ensuring that the package is user-friendly.

Both Parnika and Sherry collaborated on creating the vignettes, offering practical insights and visualizations on top authors, publication trends, and software downloads. Parnika took responsibility for writing and executing the unit tests to ensure the functions performed correctly and contributed significantly to the documentation. Sherry finalized the project by writing the README file, which provides an overview of the package, along with installation and usage instructions.

Throughout the project, Michael and Rob provided critical guidance, particularly in addressing technical issues and ensuring the package's functionality.David's continuous encouragement and input on improving the report further enhanced the quality of the final deliverable. Their input greatly contributed to the project's success.

The work was evenly distributed, with each team member contributing to different components of the project. Through collaboration and the support of their hosts, Parnika and Sherry successfully created a robust and functional tool for analyzing academic outputs at Monash EBS. The publication-scraper stands as a solid foundation for future development and extensions.

11 Future Work and Extensions - Some sort of reflection on how successful you were and possible next steps

The publication-scraper package has successfully streamlined the process of retrieving and analyzing academic outputs from ORCID, Google Scholar, and CRAN, offering a comprehensive tool for tracking research productivity at Monash EBS. However, there are several potential extensions to enhance its functionality:

-Improved Data Cleaning: While the package effectively retrieves publication data, there are instances of missing or incomplete author information. Future iterations could include more robust cleaning functions to handle missing data more effectively, possibly utilizing external databases to fill in gaps.

- Automation of Dataset Updates: Automating the updating of ORCID and Google Scholar IDs would further streamline the process and ensure the dataset remains up-to-date without manual intervention.
- Cross-Institutional Comparison: Expanding the package to handle data from multiple institutions could make it possible to perform benchmarking and cross-institutional comparisons. This would be especially useful for academic departments wanting to compare their performance with peer institutions, fostering a competitive research environment.
- Interactive Dashboards: Developing an interactive dashboard using R packages like Shiny would provide a more dynamic user experience. Users could filter data, explore trends, and generate custom reports in real-time, making the analysis process more intuitive and accessible, especially for non-technical users like administrators.

12 References

13 Appendix(if required)