BiblioMetric-Analyzer: Analysis of the Scientometrics of famous scientific authors

Diogo Abreu[a], João Machado[a], Pedro Lopes[a]

[a] *University of Minho, Braga, Portugal*

Abstract

Assessing the long-term scientific impact of researchers is a complex task, traditionally dominated by citation-based metrics such as the h-index. While widely adopted, the h-index can be artificially inflated by large publication volumes and provides limited insight into the structural evolution of a researcher's career. This work proposes a longitudinal framework for analyzing the year-by-year trajectory of the h-index and extracting shape-based parameters that capture the stability and growth dynamics of scientific impact. Using automated data acquisition from the Scopus API, we construct historical h-index curves for selected authors and fit multiple growth models, including a Hirsch-inspired square-root model. From these fits, we derive two novel indicators: the Hirsch Growth Coefficient (HGC), which measures intrinsic impact-accumulation speed, and the Stability-adjusted Hirsch Index (SaH), which penalizes instability in citation growth. This study contributes an alternative, model-based view of scientific influence and lays the groundwork for predictive bibliometrics.

Keywords: Bibliometrics, H-Index, Scientific Impact Measurement, Citation Analysis, Longitudinal Researcher Evaluation, Impact Growth Modeling, Scientometrics

Code metadata

Current code version	v1.0
Permanent link to code/repository used for this code version	https://github.com/pinetreeaxe/BiblioMetric-Analyzer
Permanent link to Reproducible Capsule	
Legal Code License	MIT License
Code versioning system used	git
Software code languages, tools, and services used	Python 3.8+, Scopus API, Streamlit, pandas, plotly, scikit-learn, scipy, python-dotenv, numpy, requests
Compilation requirements, operating environments & dependencies	Python 3.8+, pip install -r requirements.txt; tested on Linux/macOS/Windows
If available, link to developer documentation/manual	https://github.com/pinetreeaxe/BiblioMetric-Analyzer/blob/main/README.md
Support email for questions	

1. Introduction

The software presented in this article is a research tool designed to automatically extract, process, and analyze longitudinal scientific impact metrics for individual researchers. Using the Scopus API as its primary data source [1], the tool collects complete publication records, citation histories, and metadata, allowing it to reconstruct the annual evolution of the h-index and related indicators for each author.

The system aims to support researchers in scientometrics, research evaluation, and the study of scientific careers. While traditional bibliometric platforms often provide only static snapshots of citation metrics, our software enables a fully dynamic, year-by-year reconstruction of impact trajectories. This allows users to explore the stability, growth patterns, and structural behavior of metrics such as the h-index, the g-index, and the m-index, as well as new shape-based indicators introduced in this work [2].

By offering an automated pipeline, from data extraction to visual analytics, the tool makes advanced longitudinal bibliometric analysis accessible to researchers with no programming or data-engineering expertise. The implementation includes both a command-line interface (CLI) for batch processing and an interactive Streamlit web dashboard for exploratory analysis.

2. Software Description

2.1 Software Architecture

The code base is organized into seven core Python modules under the `src/` directory: `api_client.py` (Scopus API interaction), `data_processing.py` (data cleaning and structuring), `metrics.py` (standard bibliometric indicators), `shape_metrics.py` (novel shape-based parameters), `prediction_models.py` (growth curve fitting), `dashboard.py` (Streamlit interface), and `cli.py` (command-line interface). A `quota_status.json` file tracks API usage to respect rate limits. Figure 1 illustrates the complete workflow.

2.1.1 Scopus API Client (`api_client.py`)

The `api_client.py` module handles authentication via API keys stored in a `.env` file and implements robust querying of Scopus endpoints. [3] It uses the Search API to retrieve complete publication lists by Scopus Author ID (AU-ID) and the Abstract Retrieval API for detailed metadata including citation counts, DOIs, publication years, co-authors, journal details, and open-access status. Pagination, rate limiting, and error retry logic ensure complete data collection even for prolific authors.

2.1.2 Data Processing (`data_processing.py`)

Raw API responses are parsed, cleaned, and normalized in `data_processing.py`. This module handles date harmonization, duplicate removal, missing value imputation, and data validation. It constructs time-series datasets for citation evolution and exports processed data as CSV and JSON files with comprehensive metadata. The module also generates intermediate files for resuming interrupted analyses of authors with thousands of publications.

2.1.3 Standard Metrics (`metrics.py`)

The `metrics.py` module computes established indicators including the year-by-year h-index, g-index, m-index (h-index divided by career length), and i10-index [4]. These metrics are calculated across publication subsets and time windows, enabling sensitivity analysis and temporal benchmarking.

2.1.4 Shape-Based Metrics (`shape_metrics.py`)

Extending standard metrics, `shape_metrics.py` derives novel indicators: the Hirsch Growth Coefficient (HGC) from curve slope analysis, the Stability-adjusted Hirsch Index (SaH) which penalizes growth volatility, and coefficients of variation for h-index increments. These capture trajectory stability and intrinsic growth rates beyond total citation volume [5].

2.1.5 Prediction Models (`prediction_models.py`)

The `prediction_models.py` module fits multiple regression models to h-index time series: linear, polynomial, exponential, Hirsch-inspired square-root ($h(t) \propto \sqrt{t}$), and power-law models [4]. Model diagnostics (R^2, residuals, AIC) identify the best-fitting growth pattern for each researcher, supporting trajectory forecasting.

2.1.6 Command-Line Interface (`cli.py`)

`cli.py` provides a menu-driven CLI for batch processing multiple authors. Users input AU-IDs via command line or file, select analysis options (metrics, models, exports), and monitor progress. It supports job pausing/resuming and generates summary reports.

2.1.7 Interactive Dashboard (`dashboard.py`)

The Streamlit app in `dashboard.py` offers dynamic visualizations: publication timelines, citation distributions, h-index curves with overlaid model fits, metric heatmaps, and interactive parameter explorers. Users input AU-IDs via a sidebar form to trigger on-demand analysis.

3. Impact Overview

The software provides an integrated environment for the longitudinal analysis of scientific impact at the level of individual researchers. By reconstructing the year-by-year trajectory of the h-index and related indicators, it enables forms of analysis that are not directly supported by traditional bibliometric platforms, which typically expose only static snapshots of citation metrics [6].

3.1 New research questions enabled

By automating the reconstruction of longitudinal impact curves, the software makes it possible to investigate research questions such as:

- How stable is the growth pattern of the h-index over a researcher's career?
- Are there characteristic impact trajectories associated with highly distinguished researchers?
- Can early-career h-index dynamics be used as weak predictors of later high-impact recognition, such as major awards? [7]
- How do alternative shape-based metrics compare to the h-index in terms of robustness to high publication volume?

Without automated extraction and processing of citation data over time, these questions would be difficult to address at scale.

3.2 Improvements over existing workflows

In many practical settings, the analysis of a researcher's impact requires manual collection of publication lists, citation counts, and ad hoc computation of metrics, often using a combination of web interfaces and spreadsheet tools. This process is time-consuming, error-prone, and hard to reproduce.

The proposed software improves this workflow by:

- Automating complete data acquisition from the Scopus API for any given author via CLI or web interface;

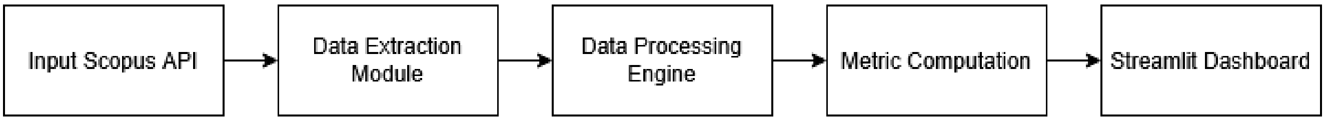

Figure 1: Pipeline of the data extraction and analysis system.

- Reconstructing historical h-index curves with model-based forecasting in a consistent manner;
- Computing both standard and novel shape-based metrics from unified datasets;
- Providing interactive Streamlit visualizations alongside exportable CSV/JSON results for non-programmers.

As a result, longitudinal bibliometric analysis that previously required substantial manual effort can now be performed in a matter of seconds per author.

3.3 Example usage scenarios

The tool can be used by scientometrics researchers studying citation dynamics, by supervisors monitoring the publication progress of students, by research evaluation committees requiring a more detailed view of scientific impact, and by students learning about bibliometric indicators. The dual CLI/dashboard interface supports both batch processing of author cohorts and interactive exploration of individual trajectories.

4. Conclusion

This paper presented a software tool for the automated extraction, processing and analysis of longitudinal citation data at the level of individual researchers. By reconstructing the yearly evolution of the h-index and fitting simple growth models, the tool supports a richer and more transparent assessment of scientific careers than static bibliometric indicators alone.

The introduction of shape-based parameters such as the Hirsch Growth Coefficient and the Stability-adjusted Hirsch index provides an alternative, model-driven perspective on scientific impact, highlighting differences in the speed and stability of h-index growth. Combined with an interactive dashboard, the software lowers the barrier for conducting longitudinal bibliometric studies and opens the door to new research questions in predictive scientometrics.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT and Grammarly to enhance their English grammar writing. After using this tool/service, the authors reviewed and edited the content as needed and took full responsibility for the content of the publication.

Acknowledgments

The authors would like to thank Prof. João Miguel Fernandes for valuable feedback on the design of the software and the interpretation of the bibliometric results.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] pybliometrics developers, pybliometrics: Python-based api-wrapper to access scopus and sciencedirect, https://pybliometrics.readthedocs.io, accessed: 2025-11-29 (2024).

[2] M. Pennisi, other authors, A new bibliometric index based on the shape of the citation distribution, PLoS ONE 9 (12) (2014) e116062. doi:10.1371/journal.pone.0115962.

[3] Elsevier Developer Portal, Elsevier developer portal: Technical documentation, https://dev.elsevier.com/technical_documentation.html (2025).

[4] J. E. Hirsch, An index to quantify an individual's scientific research output, Proceedings of the National Academy of Sciences 102 (46) (2005) 16569–16572. doi:10.1073/pnas.0507655102.

[5] M. Olensky, H-index sequences across fields: A comparative analysis, WWW Companion Proceedings (2016) 407–412doi:10.1145/2872518.2889373.

[6] S. Alonso, F. J. Cabrerizo, E. Herrera-Viedma, F. Herrera, h-index: A review focused in its variants, computation and standardization for personal research evaluation, Computer Communications 32 (1) (2009) 14–18. doi:10.1016/j.comcom.2008.09.015.

[7] L. Bornmann, L. Leydesdorff, The h-index for countries and the prediction of research performance, Scientometrics 88 (3) (2011) 951–966.