12 Author-Level Analysis and Disambiguation

R. Heller · Date: 2026-05-09 · Last changed: 2026-10-31

12.1 Learning objectives

After completing this chapter, you will be able to:

- Explain the author disambiguation problem and its impact on bibliometric analysis
- Retrieve and interpret OpenAlex author entities, including name variants
- Integrate ORCID identifiers to validate author identity
- Compute author-level productivity, citation, and topical statistics from works data
- Build a reproducible author profile report from OpenAlex

12.2 Setup

```
library(tidyverse)
library(openalexR)
library(glue)
library(gt)

set.seed(20260509)

source(here::here("R", "api_helpers.R"))
source(here::here("R", "utils.R"))
source(here::here("R", "sci_palette.R"))
```

12.3 Conceptual background

Correctly attributing publications to individual researchers is a prerequisite for any author-level analysis, yet it remains one of the hardest problems in bibliometrics. **Author disambiguation** — determining whether two name strings refer to the same person — is complicated by name variants (initials vs. full names, transliterations, name changes after marriage), homonymy (different people sharing a name), and inconsistent metadata across databases.

OpenAlex (Priem et al. 2022) addresses this through algorithmic author clustering that groups works likely belonging to the same person. Each cluster receives a unique author ID (e.g., `A5023888391`). The algorithm considers name similarity, co-author overlap, institutional affiliation, topic continuity, and citation patterns. While imperfect, this clustering is transparent and continuously improved.

ORCID (Open Researcher and Contributor ID) provides a complementary solution: a persistent digital identifier that researchers claim and curate themselves. When an ORCID is linked to an OpenAlex author entity, it provides strong confirmation of identity. However,

ORCID adoption is uneven across disciplines and geographies, so it cannot serve as the sole disambiguation strategy.

The **h-index** (Hirsch 2005) and related indicators (10.3) are commonly used to summarise author-level impact, but they are meaningful only when the underlying publication set is correctly attributed. A split author (one person assigned two IDs) will have a deflated h-index; a merged author (two people sharing one ID) will have an inflated one.

Author-level analysis also raises ethical concerns. The Leiden Manifesto (Hicks et al. 2015) warns against reducing a researcher's contribution to a single number. Career breaks, discipline-specific norms, and the Matthew effect (Merton 1968) — where established researchers accumulate citations disproportionately — must all be considered when interpreting author-level metrics.

12.4 Worked example

12.4.1 Retrieving an author entity

We retrieve the OpenAlex profile for a well-known scientometrician.

```
author_entity ← fetch_openalex(
  entity = "authors",
  search = "Ludo Waltman",
  options = list(sort = "cited_by_count:desc")
)

author_info ← author_entity ▷ slice(1)
cat(glue("Name: {author_info$display_name}\n"))
```

```
#> Name: Ludo Waltman
```

```
cat(glue("OpenAlex ID: {author_info$id}\n"))
```

```
#> OpenAlex ID: https://openalex.org/A5027467000
```

```
cat(glue("ORCID: {author_info$orcid}\n"))
```

```
#> ORCID: https://orcid.org/0000-0001-8249-1752
```

```
cat(glue("Works count: {author_info$works_count}\n"))
```

```
#> Works count: 478
```

```
cat(glue("Cited by count: {author_info$cited_by_count}\n"))
```

```
#> Cited by count: 68387
```

12.4.2 Fetching the author's works

```
author_works ← fetch_openalex(
  entity = "works",
  author.id = author_info$id,
  from_publication_date = "2010-01-01",
  to_publication_date = "2023-12-31"
)

works_slim ← author_works ▷
  select(id, display_name, publication_date, cited_by_count, type,
source_display_name) ▷
  mutate(year = year(publication_date))

cat(glue("Works retrieved: {nrow(works_slim)}\n"))
```

```
#> Works retrieved: 207
```

12.4.3 Author-level statistics

```
h_idx ← compute_h_index(works_slim$cited_by_count)

stats ← tibble(
  Metric = c("Total works", "Total citations", "h-index",
             "Mean citations/paper", "Median citations/paper",
             "First publication year", "Most recent year"),
  Value = c(
    nrow(works_slim),
    sum(works_slim$cited_by_count),
    h_idx,
    round(mean(works_slim$cited_by_count), 1),
    median(works_slim$cited_by_count),
    min(works_slim$year),
    max(works_slim$year)
  )
)

stats ▷ gt()
```

Metric	Value
Total works	207
Total citations	35061
h-index	54
Mean citations/paper	169

Metric	Value
Median citations/paper	7
First publication year	2010
Most recent year	2023

12.4.4 **Publication timeline**

```
annual ← works_slim ▷
  group_by(year) ▷
  summarise(n_pubs = n(), cites = sum(cited_by_count), .groups = "drop") ▷
  mutate(cum_cites = cumsum(cites))

ggplot(annual, aes(x = year)) +
  geom_col(aes(y = n_pubs), fill = palette_sci(1), width = 0.7) +
  geom_line(aes(y = cum_cites / max(cum_cites) * max(n_pubs)),
            colour = palette_sci(2)[2], linewidth = 1) +
  scale_y_continuous(
    name = "Publications per year",
    sec.axis = sec_axis(~ . / max(annual$n_pubs) * max(annual$cum_cites),
                        name = "Cumulative citations")
  ) +
  labs(x = "Year") +
  theme_sci()
```

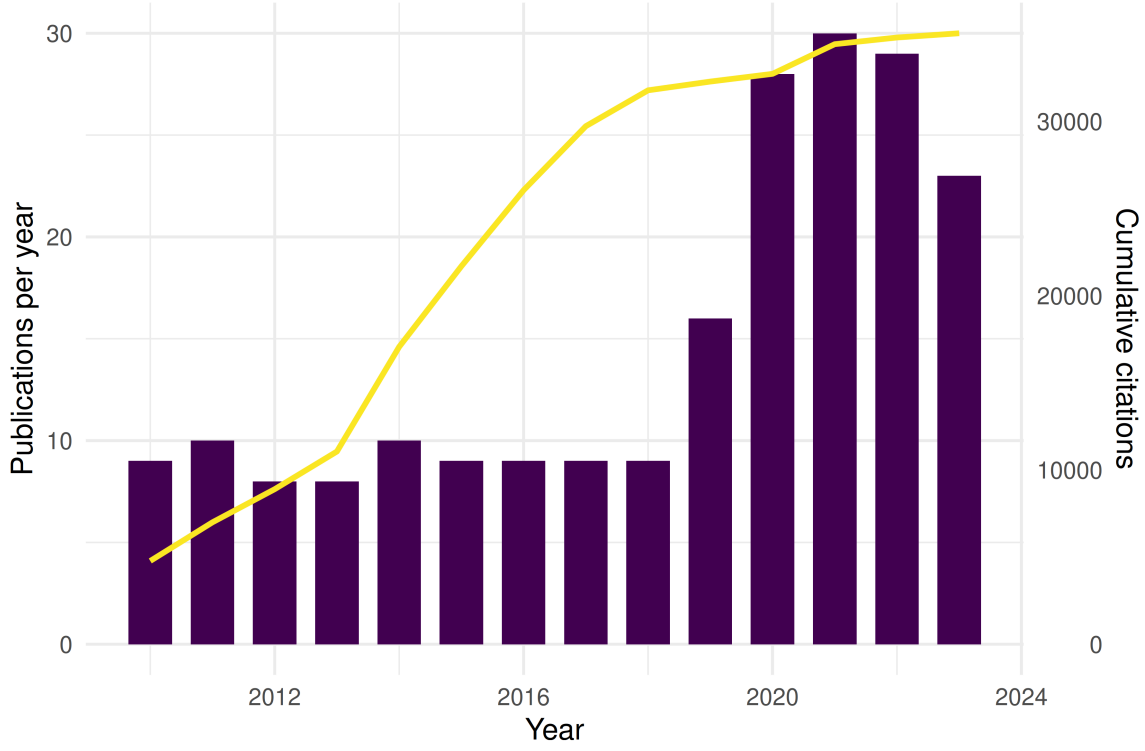


Figure 12.1: Annual publication output and cumulative citation count.

12.4.5 Top venues

```
venue_counts ← works_slim ▷
  filter(!is.na(source_display_name)) ▷
  count(source_display_name, sort = TRUE) ▷
  head(10)

ggplot(venue_counts, aes(x = n, y = reorder(source_display_name, n))) +
  geom_col(fill = palette_sci(1)) +
  labs(x = "Number of articles", y = NULL) +
  theme_sci()
```

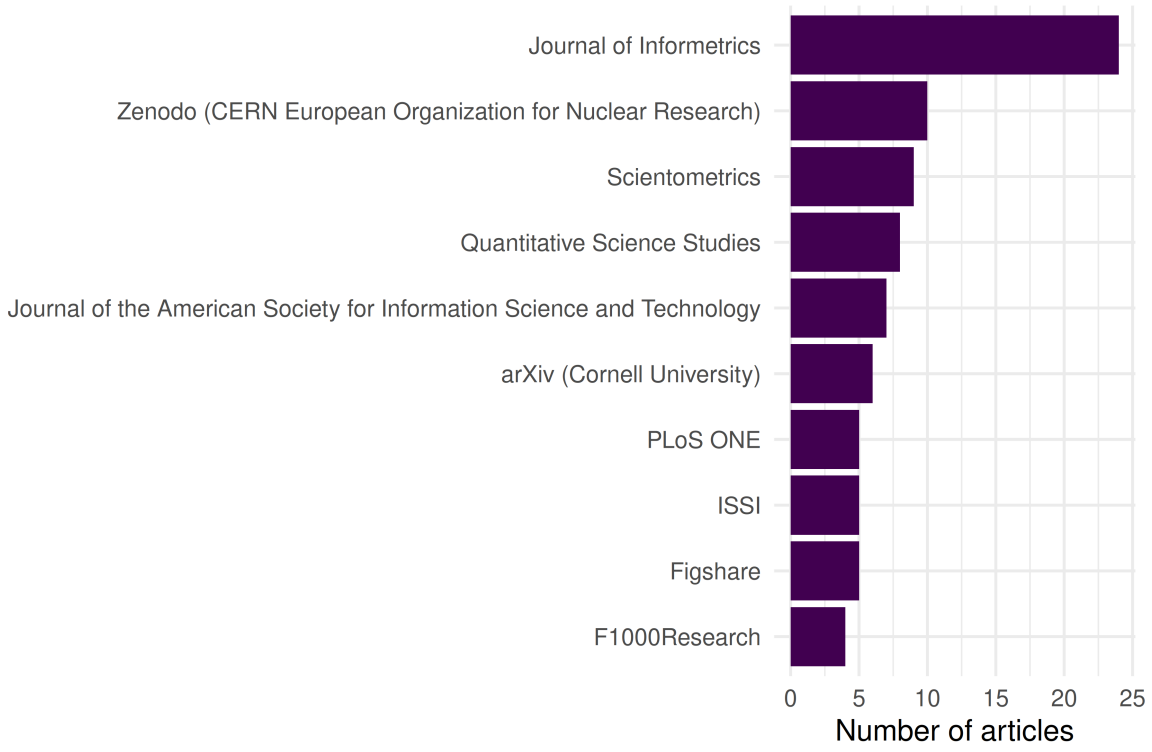


Figure 12.2: Top 10 venues by number of publications for this author.

12.4.6 Checking for name variants

```
if ("display_name_alternatives" %in% names(author_info)) {
  cat("Known name variants:\n")
  print(author_info$display_name_alternatives)
} else {
  cat("No alternative display names recorded in this entity.\n")
}
```

```
#> Known name variants:
#> [[1]]
#>  [1] "L Waltman"             "L. Waltman"             "LUDO WALTMAN"
#>  [4] "Ludo Waltman"          "Ludo Waltman, Waltman" "Ludo Waltmann"
#>  [7] "Waltman L."            "Waltman, L."            "Waltman, L. (Ludo)"
#> [10] "Waltman, L.R."         "Waltman, LR (Ludo)"    "Waltman, Ludo"
#> [13] "Waltmann, Ludo"
```

12.5 Diagnostics and interpretation

When building author profiles, verify the following:

- **Works count plausibility**: Does the number of retrieved works match the author's known output? Large discrepancies suggest a split or merged entity.
- **Topical coherence**: Scan titles and venues. If the works span wildly unrelated fields, the entity may conflate two researchers with similar names.
- **ORCID match**: If the author has an ORCID, cross-check that the OpenAlex entity links to it.
- **Co-author consistency**: The author's frequent co-authors should be recognisable collaborators, not strangers from an unrelated field.
- **Citation distribution**: An extreme outlier (e.g., one paper with 10,000 citations while all others have fewer than 50) may indicate an attribution error.

12.6 Limitations and responsible use

12.7 Limitations and responsible use

- **Disambiguation is imperfect.** OpenAlex's algorithm will sometimes split one person into multiple entities or merge distinct individuals. Always validate profiles manually for high-stakes analyses.
- **ORCID coverage is uneven.** Researchers in the Global South, early-career scholars, and those in humanities adopt ORCID at lower rates. Filtering to ORCID-linked authors introduces bias.
- **The Matthew effect.** Established researchers accumulate citations disproportionately (Merton 1968). Author-level metrics favour seniority and visibility, not necessarily quality or originality.
- **Career breaks and part-time research.** Indicators like the h-index penalise researchers who take parental leave, work part-time, or switch careers. The m-quotient is only a rough correction (Hirsch 2005).
- **Never evaluate individuals by numbers alone** (Hicks et al. 2015; American Society for Cell Biology 2012). Author profiles should supplement, not replace, reading the actual work.

12.8 Common pitfalls

12.9 Common pitfalls

- **Trusting a single database.** OpenAlex may lack works indexed only in domain-specific databases (e.g., SSRN, arXiv preprints not yet linked). Cross-check with ORCID profiles.
- **Ignoring co-author contributions.** Raw publication counts treat single-author and 50-author papers equally. Consider fractional counting for productivity analysis.
- **Comparing across career stages.** An early-career researcher with h = 8 after 5 years may be more productive than a senior researcher with h = 25 after 30 years. Use the m-quotient or age-normalised indicators.
- **Conflating author-level and paper-level metrics.** An author's h-index says nothing about which of their papers are excellent and which are not.

12.10 Exercises

1. **Profile comparison.** Fetch profiles for two researchers in the same field. Compare their h-index, publication count, and mean citations per paper. What does each metric reveal that the others do not?
2. **ORCID validation.** For an author with an ORCID, fetch their works via both OpenAlex author ID and ORCID filtering. Do the two sets agree? What might explain discrepancies?
3. **Name variant detection.** Search OpenAlex for a common name (e.g., "Wang Wei"). How many author entities are returned? What heuristics could you use to identify the correct one?
4. **Fractional productivity.** Recompute annual publication counts using fractional counting (1/k credit per paper with k authors). How does this change the author's productivity curve?
5. **Self-citation ratio.** For an author's works, estimate the fraction of citations that come from the author's own other papers. *(Hint: check if cited works share the same author ID.)*

12.11 Solutions

> Solutions are provided in E.

12.12 Further reading

- Priem et al. (2022) — OpenAlex's author disambiguation and entity resolution approach.
- Hirsch (2005) — The h-index, the most widely used author-level indicator.
- Merton (1968) — The Matthew effect: cumulative advantage in scientific recognition.
- Hicks et al. (2015) — The Leiden Manifesto: principles for responsible author evaluation.
- American Society for Cell Biology (2012) — DORA: recommendations against reducing researchers to single numbers.
- Egghe (2006) — The g-index, a complement to the h-index for author evaluation.

12.13 Session info

```
sessionInfo()
```

```
#> R version 4.6.1 (2026-06-24)
#> Platform: x86_64-pc-linux-gnu
#> Running under: Ubuntu 24.04.5 LTS
#>
#> Matrix products: default
#> BLAS:   /usr/lib/x86_64-linux-gnu/openblas-pthread/libblas.so.3
#> LAPACK: /usr/lib/x86_64-linux-gnu/openblas-pthread/libopenblasp-r0.3.26.so;  LAPACK
version 3.12.0
#>
#> locale:
#>  [1] LC_CTYPE=C.UTF-8       LC_NUMERIC=C           LC_TIME=C.UTF-8
#>  [4] LC_COLLATE=C.UTF-8     LC_MONETARY=C.UTF-8    LC_MESSAGES=C.UTF-8
#>  [7] LC_PAPER=C.UTF-8       LC_NAME=C              LC_ADDRESS=C
#> [10] LC_TELEPHONE=C         LC_MEASUREMENT=C.UTF-8 LC_IDENTIFICATION=C
#>
#> time zone: UTC
#> tzcode source: system (glibc)
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#>  [1] quanteda_4.5.1     pdftools_3.9.1     arrow_26.0.1.1     bibliometrix_5.5.0
#>  [5] RefManageR_1.4.0   bib2df_1.1.2.0     rcrossref_1.2.1    gt_1.4.0
#>  [9] tidytext_0.4.3     glue_1.8.1         openalexR_3.1.2    lubridate_1.9.5
#> [13] forcats_1.0.1      stringr_1.6.0      dplyr_1.2.1        purrr_1.2.2
#> [17] readr_2.2.0        tidyr_1.3.2        tibble_3.3.1       ggplot2_4.0.4
#> [21] tidyverse_2.0.0
#>
#> loaded via a namespace (and not attached):
#>   [1] gridExtra_2.3.1        httr2_1.3.0            readxl_1.6.0
#>   [4] rlang_1.3.0            magrittr_2.0.5         otel_0.3.0
#>   [7] compiler_4.6.1         vctrs_0.7.3            httpcode_0.3.0
#>  [10] pkgconfig_2.0.3        fastmap_1.2.0          labeling_0.4.3
#>  [13] ca_0.72                utf8_1.2.6             promises_1.5.0
#>  [16] rmarkdown_2.32         tzdb_0.5.0             bit_4.6.0
#>  [19] xfun_0.62              cachem_1.1.0           jsonlite_2.0.0
#>  [22] SnowballC_0.7.1        later_1.4.8            parallel_4.6.1
#>  [25] stopwords_2.3          R6_2.6.1               bslib_0.12.0
#>  [28] stringi_1.8.9          RColorBrewer_1.1-3     cellranger_1.1.0
#>  [31] jquerylib_0.1.4        assertthat_0.2.1       Rcpp_1.1.2
#>  [34] bookdown_0.48          knitr_1.52             triebeard_0.4.1
#>  [37] base64enc_0.1-6        rentrez_1.2.4          httpuv_1.6.17
#>  [40] Matrix_1.7-5           igraph_2.3.4           timechange_0.4.0
#>  [43] tidyselect_1.2.1       stringdist_0.9.17      pubmedR_1.1.0
#>  [46] rstudioapi_0.19.0      yaml_2.3.12            viridis_0.6.5
#>  [49] miniUI_0.1.2           humaniformat_0.6.0     curl_8.1.0
#>  [52] qpdf_1.4.1             lattice_0.22-9         plyr_1.8.9
#>  [55] shiny_1.15.0           withr_3.0.3            S7_1.0.0
#>  [58] askpass_1.2.1          evaluate_1.0.5         zip_3.0.2
#>  [61] xml2_1.6.0             shinycssloaders_1.1.0  pillar_1.11.1
#>  [64] janeaustenr_1.0.0      DT_0.34.0              plotly_4.12.1
#>  [67] generics_0.1.4         rprojroot_2.1.1        hms_1.1.4
#>  [70] scales_1.4.0           xtable_1.8-8           contentanalysis_1.1.1
#>  [73] tools_4.6.1            data.table_1.18.6.1    tokenizers_0.3.0
#>  [76] openxlsx_4.2.9         XML_3.99-0.25          fs_2.1.0
#>  [79] visNetwork_2.1.4       fastmatch_1.1-8        grid_4.6.1
#>  [82] bibtex_0.5.3           urltools_1.7.3.1       rscopus_0.9.0
#>  [85] dimensionsR_0.0.3      bibliometrixData_0.3.0 cli_3.6.6
#>  [88] viridisLite_0.4.3      downlit_0.4.5          gtable_0.3.6
#>  [91] sass_0.5.0             digest_0.6.39          ggrepel_0.9.8
#>  [94] crul_1.6.0             htmlwidgets_1.6.4      farver_2.1.2
#>  [97] memoise_2.0.1          htmltools_0.5.9        lifecycle_1.0.5
```

```
#> [100] httr_1.4.9             here_1.0.2             mime_0.13
#> [103] bit64_4.8.6
```