

# Package ‘biblioverlap’

November 7, 2023

**Type** Package

**Title** Document-Level Matching Between Bibliographic Datasets

**Version** 1.0.2

**Description** Identifies and visualizes document overlap in any number of bibliographic datasets.

This package implements the identification of overlapping documents through the exact match of a unique identifier (e.g. Digital Object Identifier - DOI) and, for records where the identifier is absent, through a score calculated from a set of fields commonly found in bibliographic datasets (Title, Source, Authors and Publication Year).

Additionally, it provides functions to visualize the results of the document matching through a Venn diagram and/or UpSet plot, as well as a summary of the matching procedure.

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.2.3

**Imports** dplyr, ggplot2, ggVennDiagram, magrittr, Matrix, parallel, rlang, shiny, stringdist, UpSetR, uuid

**Suggests** DT, testthat

**Depends** R (>= 4.1)

**URL** <https://github.com/gavieira/biblioverlap>

**BugReports** <https://github.com/gavieira/biblioverlap/issues>

**NeedsCompilation** no

**Author** Gabriel Vieira [aut, cre, cph]  
(<https://orcid.org/0000-0002-5529-7628>),  
Jacqueline Leta [ctb] (<https://orcid.org/0000-0002-3271-7749>)

**Maintainer** Gabriel Vieira <[gabriel.vieira@bioqmed.ufrj.br](mailto:gabriel.vieira@bioqmed.ufrj.br)>

**Repository** CRAN

**Date/Publication** 2023-11-07 19:50:02 UTC

## R topics documented:

biblioverApp . . . . .	2
biblioverlap . . . . .	3
plot_matching_summary . . . . .	5
plot_upset . . . . .	5
plot_venn . . . . .	6
ufrj_bio_0122 . . . . .	7
<b>Index</b>	<b>8</b>

---

biblioverApp	<i>Shiny App for the biblioverlap package</i>
--------------	---

---

### Description

Shiny App for the biblioverlap package

### Usage

```
biblioverApp(port = NULL, max_upload_size = 1000, launch.browser = TRUE)
```

### Arguments

port	• port of the application
max_upload_size	• max upload size of documents (in MB) - Default 100
launch.browser	• launch on browser - Default = TRUE

### Value

opens a instance of the biblioverlap UI

### Examples

```
#Running the ShinyApp
biblioverApp()
```

---

 biblioverlap

*Document-level matching of bibliographic datasets*


---

## Description

This function identifies document overlap between bibliographic datasets and records it through the use of Universally Unique Identifiers (UUID).

## Usage

```

biblioverlap(
  db_list,
  matching_fields = default_matching_fields,
  n_threads = 1,
  ti_penalty = 0.1,
  ti_max = 0.6,
  so_penalty = 0.1,
  so_max = 0.3,
  au_penalty = 0.1,
  au_max = 0.3,
  py_max = 0.3,
  score_cutoff = 1
)

```

## Arguments

db_list	• list of dataframes containing the sets of bibliographic data
matching_fields	• Five column names used in the matching. Should be universal across all datasets and provided as a named list with the following names: <b>DI</b> (unique identifier), <b>TI</b> (document title), <b>PY</b> (publication year), <b>SO</b> (publication source) and <b>AU</b> (Authors). Default values come from <a href="#">The Lens scholar field definition</a> .
n_threads	• number of (logical) cores used in the matching procedures. Default: 1
ti_penalty	• penalty applied for each increment in Title's Levenshtein distance. Default: 0.1
ti_max	• max score value for Title. Default: 0.6
so_penalty	• penalty applied for each increment in Source's Levenshtein distance. Default: 0.1
so_max	• max score value for Source. Default: 0.3
au_penalty	• penalty applied for each increment in Author's Levenshtein distance. Default: 0.1
au_max	• max score value for Author. Default: 0.3
py_max	• max score value for Publication Year. Default: 0.3
score_cutoff	• minimum final score for a valid match between two documents. Default: 1

## Details

In this procedure, any duplicates in the same dataset are removed. Then, Universally Unique Identifiers (UUID) are attributed to each record. If a match is found between two documents in a pairwise comparison, the UUID of the record from the first dataset is copied to the record on the second.

All preprocessing and modifications to the dataset are performed in a copy of the original data, which is used internally by the program. After all pairwise comparisons are completed, the UUID data is added as a new column in the original data.

Thus, the `db_list` returned by this function contains the same fields provided by the user plus the UUID column with the overlap information. This allows for further analysis using other fields (e.g. 'number of citations' or 'document type').

## Value

a list object containing:

- (i) `db_list`: a modified version of `db_list` where matching documents share the same UUID
- (ii) `summary`: a summary of the results of the matching procedure

## Note

In its internal data, the program will attempt to split the AU (Author) field to extract only the first author, for which it will calculate the Levenshtein distance.

It assumes that the AU field is ";" (semicolon) separated. Thus, in order to correctly perform the matching procedure to when another separator is being applied to this field, the user can either: (i) change the separator to semicolon; or (ii) create a new column containing only the first author.

## Examples

```
#Example list of input dataframes
lapply(ufrj_bio_0122, head, n=1)

#List of columns for matching (identical to biblioverlap()'s defaults)
matching_cols <- list(DI = 'DOI',
                     TI = 'Title',
                     PY = 'Publication Year',
                     AU = 'Author/s',
                     SO = 'Source Title')

#Running document-level matching procedure (first two dataframes)
biblioverlap_results <- biblioverlap(ufrj_bio_0122[1:2], matching_fields = matching_cols)

#Taking a look at the matched db_list
lapply(biblioverlap_results$db_list, head, n=1)

#Taking a look at the matching results summary
biblioverlap_results$summary
```

---

plot\_matching\_summary *Plotting biblioverlap's matching summary*

---

**Description**

Plotting biblioverlap's matching summary

**Usage**

```
plot_matching_summary(matching_summary_df, ...)
```

**Arguments**

matching\_summary\_df

- summary of matching process generated by `biblioverlap()`

...

- additional arguments passed down to `ggplot2::geom_text()`

**Value**

a barplot summary of the matching results

**Examples**

```
#Running document-level matching procedure
biblioverlap_results <- biblioverlap(ufrj_bio_0122[1:2])

#Checking biblioverlap results (summary table)
biblioverlap_results$summary

#Plotting the matching summary
plot_matching_summary(biblioverlap_results$summary)
```

---

plot\_upset *Plotting UpSet plot from biblioverlap results*

---

**Description**

Plotting UpSet plot from biblioverlap results

**Usage**

```
plot_upset(db_list, ...)
```

**Arguments**

- db\_list • list of matched dataframes (with UUID column added by biblioverlap)
- ... • arguments to be passed down to `UpSetR::upset()`

**Value**

a UpSet plot representation of document overlap between the input datasets

**Examples**

```
#Running document-level matching procedure
biblioverlap_results <- biblioverlap(ufrj_bio_0122[1:2])

#Checking biblioverlap results (db_list)
lapply(biblioverlap_results$db_list, head, n=1)

#Plotting the UpSet plot
plot_upset(biblioverlap_results$db_list)
```

---

plot\_venn

*Plotting Venn Diagram from biblioverlap results*


---

**Description**

Plotting Venn Diagram from biblioverlap results

**Usage**

```
plot_venn(db_list, ...)
```

**Arguments**

- db\_list • list of matched dataframes (with UUID column added by biblioverlap)
- ... • Additional arguments that can be passed down to `ggVennDiagram::ggVennDiagram()`

**Value**

a Venn Diagram representation of document overlap between the input datasets

**Examples**

```
#Running document-level matching procedure
biblioverlap_results <- biblioverlap(ufrj_bio_0122[1:2])

#Checking biblioverlap results (db_list)
lapply(biblioverlap_results$db_list, head, n=1)
```

```
#Plotting the Venn diagram  
plot_venn(biblioverlap_results$db_list)
```

---

ufrj_bio_0122	<i>UFRJ-affiliated documents from biological sciences disciplines (January 2022)</i>
---------------	--

---

## Description

Data obtained from [The Lens Scholarly Search](#) in September 6, 2023.

The original data contained all documents from four major biological sciences fields published in the year 2022 by at least one author affiliated to the Universidade Federal do Rio de Janeiro (UFRJ). The data was then subsampled to documents published exclusively in January 2022 to reduce package size.

The biological disciplines featured in this dataset are [Biochemistry](#), [Genetics](#), [Microbiology](#) and [Zoology](#).

## Usage

```
ufrj_bio_0122
```

## Format

```
ufrj_bio_0122:
```

A named list with 4 elements. Each element is a dataframe that contains the following fields:

**Lens ID** Unique identifier given to each record in The Lens database

**DOI** Digital Object Identifier

**Title** Document title

**Publication Year** Document publication year

**Source Title** Source (e.g. journal) where the document has been published

**Author/s** Document authors

**Publication Type** Type of the document (e.g. 'journal article', 'book chapter', etc...)

**Citing Works Count** Total number of citations received by document at the time of data recovery

**Open Access Colour** Type of open access (e.g. gold, bronze, green, etc...)

## Source

<https://www.lens.org>

# Index

## \* datasets

ufrj\_bio\_0122, [7](#)

biblioverApp, [2](#)

biblioverlap, [3](#)

biblioverlap(), [5](#)

ggplot2::geom\_text(), [5](#)

ggVennDiagram::ggVennDiagram(), [6](#)

plot\_matching\_summary, [5](#)

plot\_upset, [5](#)

plot\_venn, [6](#)

ufrj\_bio\_0122, [7](#)

UpSetR::upset(), [6](#)