refnet

Auriel M. V. Fournier, Forrest Stevens, Matt Boone, Emilio M. Bruna

2018-09-22

1. Introduction

The Science of Science (SciSci) is an emerging, trans-disciplinary approach for using large and disparate data-sets to study the emergence, dissemination, and impact of scientific research (Fortunato et al. 2018). Bibliometric databases such as the Web of Science are rich sources of data for SciSci studies (Sugimoto and Lariviare 2018). In recent years the type and scope of questions addressed with data gathered from these databases has expanded tremendously (Forutnato et al. 2018). This is due in part to their expanding coverage and greater accessibility, but also because advances in computational power make it possible to analyze data-sets comprising millions of bibliographic records (e.g., Lariviare et al. 2013, Smith et al. 2014).

The rapidly increasing size of bibliometric data-sets available to researchers has exascerbated two major and persistent challenges in SciSci research. The first of these is **Author Name Disambiguation**. Correctly identifying the authors of a research product is fundamental to bibliometric research, as is the ability to correctly attribute to a given author all of their scholarly output. However, this seemingly straightforward task can often be extremely complicated, even when using the nominally high-quality data extracted from bibliometric databases (reviewed in Smalheiser and Torvik 2009). The most obvious case is when different authors have identical names, which can be quite common in some countries (Strotmann et al. 2009). However, confusion might also arise as a result of journal conventions or individual preferences for abbreviating names. For instance, one might conclude "J. C. Smith", "Jennifer C. Smith", and "J. Smith" are different authors, when in fact they are the same person. In contrast, papers by "E. Martinez" could have been written by different authors with the same last name but whose first names start with the same letter (e.g., "Enrique", "Eduardo"). Failure to disambiguate author names can seriously undermine the conclusions of some SciSci studies, but manually verifying author identity quickly becomes impractical as the number of authors or papers in a dataset increases.

The second challenge to working with large bibliometric data-sets is correctly **parsing author addresses**. The structure of author affiliations is complex and idiosyncratic, and journals differ in the information they require authors to provide and the way in which they present it. Authors may also represent affiliations in different ways on different articles. For instance, the affiliations might be written in different ways in different journals (e.g., "Dept. of Biology", "Department of Biology", "Departamento de Biologia"). The same is true of the institution's name ("UC Davis", "University of California-Davis","University of California") or the country in which it is based ("USA", "United States", "United States of America"). Researchers at academic institutions might include the one or more Centers, Institutes, Colleges, Departments, or Programs in their address, and researchers working for the same institution could be based at units in geographically disparate locations (e.g., a University of Florida researcher could be based at one of 12 statewide Research and Education Centers, five research laboratories, 67 county extension offices, or the main campus in Gainesville). Finally, affiliations are recorded in a single field of a reference's bibliographic record, despite comprising very different types of information (e.g., city, postal code, institution). In concert, these factors can make it challenging to conduct analyses for which author affiliation or location is of particular interest.

Package `refnet` helps users of the R statistical computing environment (R Core Team 2017) address these challenges. It imports and organizes the output from Web of Science searches, disambiguates author names and suggests which might need additional scrutiny, parses author addresses, and georeferences authors' institutions. It also maps author locations and coauthorship networks. Finally, the processed data-sets can be exported in tidy formats for more in-depth analyses with user-written code packages such as `revtools` (Westgate 2018) or `bibliometrix` (Aria & Cuccurullo 2017).

2. Using refnet

Appendix 1 provides guidance on downloading records from the Web of Science in the proper format for use in `refnet`. Once bibliographic records have been downloaded, the `refnet` package's tools are applied in four steps:

1. importing and tidying reference records (Section 2.1)
2. author name disambiguation and parsing of author addresses (Section 2.2)
3. georeferencing of author institutions (Section 2.3)
4. data visualization (Section 2.4)

These examples below use the sample dataset 'example_data.txt' included with the `refnet` package. *The examples assume one has created two folders in ones working directory or Rstudio Project Folder: one named 'data' and one names 'output' - and that the exported bibliographic records are saved in the 'data' folder.*

2.1. Importing Search Results

The `refnet` package can either import a single Web of Science search result file or combine and import multiple files located in the same directory. The acceptable file formats are '.txt' and '.ciw'. Importing reference records is done with the `references_read()` function, which has three arguments:

- **data**: The location of the directory in which the Web of Science file(s) are located. If left blank it assumes the files are in the working directory. If in a different directory (e.g., the 'data' folder in the working directory), the absolute file name or relative file paths can be used.
- **dir**: TRUE when loading multiple files; FALSE when loading a single file. When multiple files are processed `refnet` identifies and removes any duplicate reference records.
- **filename_root**: The location in which the output file is to be saved (e.g., the 'output' folder in the working directory) and the prefix used to name it. If you do not want to write a file leave this field blank.

The output of `references_read()` is an object in the R Workspace and a .csv file. Each line of the output is a reference; the columns are the name of the .txt file from which the data were extracted, a unique id number assigned by `refnet` to each article, and the data from each field of the reference record (identified by the Web of Science and RIS codes for different data types (Appendix 2). This file is used by `refnet` for Step 2.

Example

a. To import and process a single file located in a folder named "data" and save the output as a file named "example_references" in the "output" folder:

```
example_refs <- references_read(data = './data/example_data.txr',
                                dir = F,
                                filename_root = './output/example')
```

b. To import and process multiple files located in a folder named "data" and save the output as "WOS_references" in the "output" folder:

```
example_refs <- references_read(data = './data',
                                dir = T,
                                filename_root = './output/example')
```

c. To save to a location with a prefix, write the folder path followed by a '/' and then the prefix. This will save the file under the root directory as "/newpath/newprefix_references.csv":

`"./newpath/newprefix".`

2.2. Author address parsing and name disambiguation

The next step is to identify all unique authors in the dataset and parse their affiliations for each of their articles. This requires identifying any authors whose name appears to be represented in different ways on different publications. Name disambiguation is a complex statistical and computational problem for which researchers have used data ranging from author affiliation to patterns of coauthorship and citation (reviewed in Smalheiser & Torvik 2009). The `authors_clean()` disambiguation algorithm, described in greater detail in Appendix 3, first assigns each author of each article a unique ID number, then assigns putative name variants representing the same author a group ID number. The function `authors_clean()` has two arguments:

- **references**: The object created by `references_read()`, from which author names will be extracted. Any previously generated output from `references_read()` that has been saved to an object can be used.
- **filename_root**: The location in which the output file is to be saved and the prefix used to name it. If you do not want to write a file leave this field blank. The syntax is the same as for `references_read()`.

The output of `authors_clean()` is a list in the R workspace with two elements: (1) "prelim", which is the initial list of disambiguated author names, and (2) "review", which is the subset of authors with putative name variants suggested for verification. Each of these elements is also saved as a separate .csv file in the location specified in the function (e.g., the output folder in the working directory).

Once disambiguation is complete, users can accept `refnet`'s results without reviewing names flagged for manual inspection (Section 2.2.1). Alternatively, users can review the subset of names recommended for inspection and make corrections if needed (Section 2.2.2). These corrections are then used to generate the 'refined' dataset used for analyses (Section 2.2.3).

Example

a. To disambiguate the authors of the references in the dataset and place the "preliminary" and "review" author lists in the "output" folder:

```
example_a_clean <- authors_clean(example_refs,
                          filename_root="./output/example")
```

2.2.1. Accepting the results of author disambiguation *without* manual review

Accepting the result of `refnet`'s disambiguation algorithm without inspecting names flagged for review is done with the `authors_refine()` function. It has four arguments:

- **review**: The names proposed for manual review by `authors_clean()`. Must be in an object.
- **prelim**: The preliminary file of disambiguated author names created with `authors_clean()`. Must be an object.
- **sim_score**: The threshold for the similarity score below which authors are assigned different groupIDs (range: 0-1; 1 = names must be identical to be assigned to the same group). By default this is turned off.
- **filename_root**: The prefix for the resulting output and locations to which it will be saved, using the same syntax as for previous functions.

The output of `authors_refine()` is an object in the R Workspace and a .csv file.

Example

a. To accept the results of author disambiguation without manual review:

```
example_a_refined <- authors_refine(example_a_clean$review,
                                    example_a_clean$prelim,
                                  filename_root="./output/example")
```

2.2.2. Reviewing and correcting the results of disambiguation

Users that prefer to manually review the results of the disambiguation can do so with the "authors" object and .csv files. A more thorough overview of the information provided in these files and how to use it to review and correct author name assignments is provided in Appendix 2).

- **If different authors were incorrectly assigned the same groupID number:** replace the number in the **groupID** column of the unique author with the value from that person's **authorID** column. *Be sure to use the authorID value from the same row.*

- **If the same author was incorrectly assigned different groupID numbers:** replace the number in the **groupID** column of the name variants to be pooled with a single **AuthorID** number. *We recommend using the lowest authorID number of the name variants being pooled.*

Example

**IMAGES MISSING 1) Image of the reivew file:

2) Same image indicating author grouped incorrectly

3) Same image name not grouped that should have been

4) Image showing final refined with corrections and instructions to same as "—-_authors_corrected.csv" in the output folder**

2.2.3. Uploading and merging the results of disambiguation

Corrections made to the "review" file are merged into the "preview" file using the `authors_refine()` function. It has four arguments:

- **corrected**: The corrected version of the "review" object. Must be an object. So read in the the corrected .csv file before using this function.
- **prelim**: The preliminary list of disambiguated author names created with `authors_clean()`. Must be in an object.
- **sim_score**: The threshold for the similarity score below which authors are assigned different groupIDs (range: 0-1; 1 = names must be identical to be assigned to the same group). By default this is turned off.
- **filename_root**: The prefix for the resulting output and locations to which it will be saved, using the same syntax as for previous functions.

The output of `authors_refine()` is a file is an object in the R Workspace and a .csv file.

Example

```
example_a_corrected <- read.csv("correctedfile.csv")

example_a_refined <-authors_refine(example_a_corrected,
                                   example_a_clean$prelim,
                                   filename_root="./output/example")
```

	A	B	C	D	E	F	
1	authorID	AU	AF	groupID	match_name	similarity	address
2	1	Cassemiro, FAS	Cassemiro, Fernanda A. S.	1	NA	NA	Univ Fed Goias, Dept Ecol, BR-74001970 Goiania
3	3160	Cassemiro, FAS	Cassemiro, F. A. S.	1	Cassemiro, Fernanda A. S.	0.92632	Univ Brasilia, Grad Program Anim Biol, Dept Zoo
4	6	Sousa-Souto, L	Sousa-Souto, L.	6	NA	NA	Univ Fed Sergipe, Grad Programme Ecol, BR-491
5	2516	Sousa-Souto, L	Sousa-Souto, Leandro	6	Sousa-Souto, L.	0.94	Univ Fed Vicosa, Dept Biol Anim, BR-36570000 V
6	3468	Sousa-Souto, L	Sousa-Souto, Leandro	6	Sousa-Souto, L.	0.94	NA
7	7	Teodoro, AV	Teodoro, A. V.	7	NA	NA	Univ Gottingen, D-37073 Gottingen, Germany.
8	2055	Teodoro, AV	Teodoro, Adenir V.	7	Teodoro, A. V.	0.92857	Univ Gottingen, D-37073 Gottingen, Germany.
9	8	Tscharntke, T	Tscharntke, T.	8	NA	NA	Univ Gottingen, D-37073 Gottingen, Germany.
10	2056	Tscharntke, T	Tscharntke, Teja	8	Tscharntke, T.	0.9625	Univ Gottingen, D-37073 Gottingen, Germany.
11	7884	Tscharntke, T	Tscharntke, Teja	8	Tscharntke, T.	0.9625	Georg August Univ Gottingen, Agroecol, Gottinge
12	9744	Tscharntke, T	Tscharntke, Teja	8	Tscharntke, T.	0.9625	Univ Gottingen, Inst Agroecol, D-37073 Gottinge
13	11980	Tscharntke, T	Tscharntke, Teja	8	Tscharntke, T.	0.9625	Univ Gottingen, Inst Agroecol, D-37073 Gottinge
14	11	Batalha, MA	Batalha, Marco A.	11	NA	NA	Univ Fed Sao Carlos, Dept Bot, BR-13565905 Sao
15	2292	Batalha, MA	Batalha, Marco Antonio	11	Batalha, Marco A.	0.93684	Univ Fed Sao Carlos, Dept Bot, BR-13565905 Sao
16	2322	Batalha, MA	Batalha, M. A.	11	Batalha, Marco A.	0.93846	Univ Fed Sao Carlos, Dept Bot, BR-13565905 Sao
17	4312	Batalha, MA	Batalha, Marco Antonio	11	Batalha, Marco A.	0.93684	NA
18	12	Silva, IA	Silva, Igor A.	12	NA	NA	Univ Fed Sao Carlos, Dept Bot, BR-13565905 Sao
19	2293	Silva, IA	Silva, Igor Aurelio	12	Silva, Igor A.	0.925	Univ Fed Sao Carlos, Dept Bot, BR-13565905 Sao
20	17	Aguiar, AV	Aguiar, Antonio V.	17	NA	NA	Univ Florida, Dept Wildlife Ecol & Conservat, Ga
21	212	Aguiar, AV	Aguiar, Antonio Venceslau	17	Aguiar, Antonio V.	0.92727	Univ Florida, Dept Wildlife Ecol & Conservat, Ga
22	892	Aguiar, AV	Aguiar, Antonio Venceslau	17	Aguiar, Antonio V.	0.92727	Univ Fed Pernambuco, Dept Bot, Programa Posg
23	18	Girao, LC	Girao, Luciana C.	18	NA	NA	Univ Fed Pernambuco, Dept Bot, Programa Posg
24	1878	Girao, LC	Girao, Luciana Coe	18	Girao, Luciana C.	0.97333	Univ Fed Pernambuco, Dept Bot, Programa Posg

2.3. Georeferencing author institutions

Users can georeference author's institutions (latitude & longitude) using the `authors_georef()` function. This function has has 3 arguments:

- **data**: The output created by `authors_refine()`. Must be an object.
- **address_column**: A quoted character identifying the column name in which addresses are stored.
- **write_out_missing**: if TRUE creates a file with the author addresses that that `refnet` was unable to georeference; set to TRUE by default

The outputs of `authors_georef()` are (1) a modified data.frame with new columns for the latitude and longitude of each authors institution and the parts of the author's address that could be parsed out from the WOS record, and (2) an output/file of references that refnet was unable to georeference, which the user can review, manually correct, and import back into the file of georeferenced author locations. Output is saved as `filename_addresses.csv`.

- **Warning #1**: it can be difficult for this version of `refnet` (v1.0) to differentiate between geographically distinct installations of the same institution (e.g. Mississippi State University Main Campus in Starkville, MS vs Mississippi State University Coastal Research and Extension in Biloxi, MS (250 miles apart)).
- **Warning #2** The `authors_georef()`function parses addresses from the Web of Science reference sheet and then attempts to calculate the latitude and longitude for them with the http://www.datasciencetoolkit.org/ or https://developers.google.com/maps/documentation/. The time required to do so depends on the number of addresses being processed and the speed of the internet connection.

Example

```
example_georef <-authors_georef(------,------,-----)
```

2.4. Data Visualization: Productivity and Collaboration

`refnet` can generate five visualizations of productivity and collaboration. **World plots** indicate the locations of authors on a map of the world, while **Net plots** are visualizations of co-authorship networks. `plot_addresses_country()` uses the `rworldmap` package, `plot_addresses_points()`, `plot_net_address()`, and `plot_net_country()` use the `ggplot2` package, and `plot_net_coauthor()` uses package `igraph`. Advanced users

familiar with these packages can customize the visualizations to suit their needs. **Warning**: The time required to render these plots is highly dependent on the number of authors in the dataset and the processing power of the computer on which analyses are being carried out.

2.4.1. World Plot 1: Authors By Country.

The `plot_addresses_country()` makes a plot whose shading indicates the number of papers with an author based in a given country. There is no fractional authorship, e.g., if an author based in the USA has authored or coauthored 3 papers in the dataset, then the USA will be credited with 3 articles

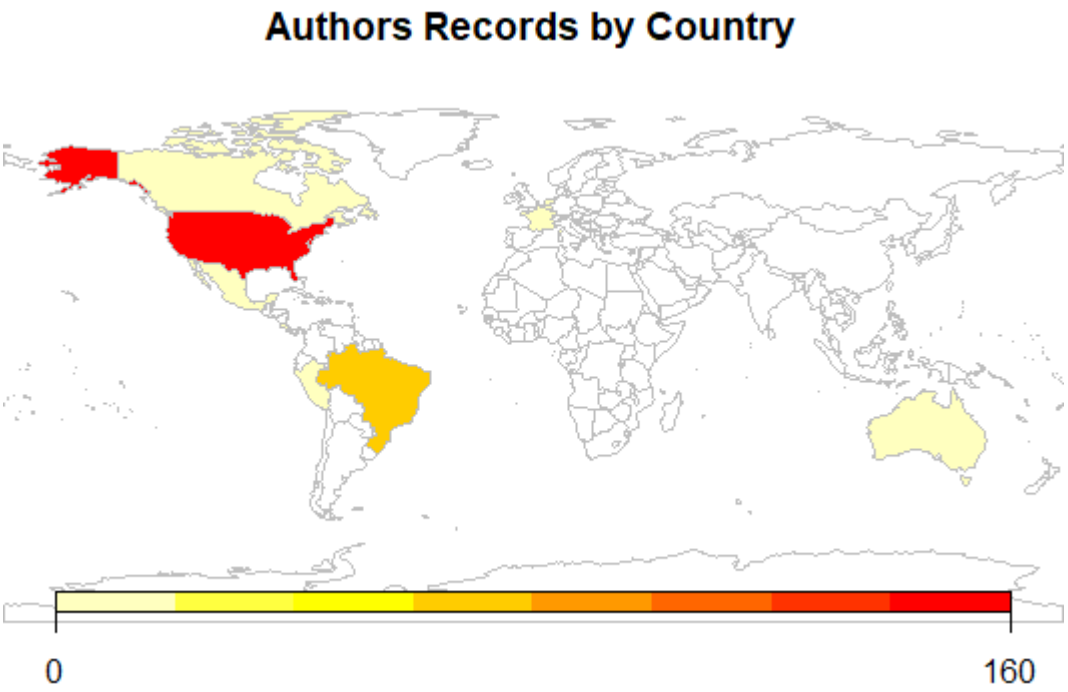
The function has one argument:

- **data**: This is the output from `authors_georef()`

The output of `plot_addresses_country()` is plot from the `rworldmap` package.

Example

```
plot_addresses_country <- plot_addresses_country(data, filename_root="./output/example")
```



2.4.2. World Plots (points)

The `plot_addresses_points()` function plots the location of all authors in the dataset.

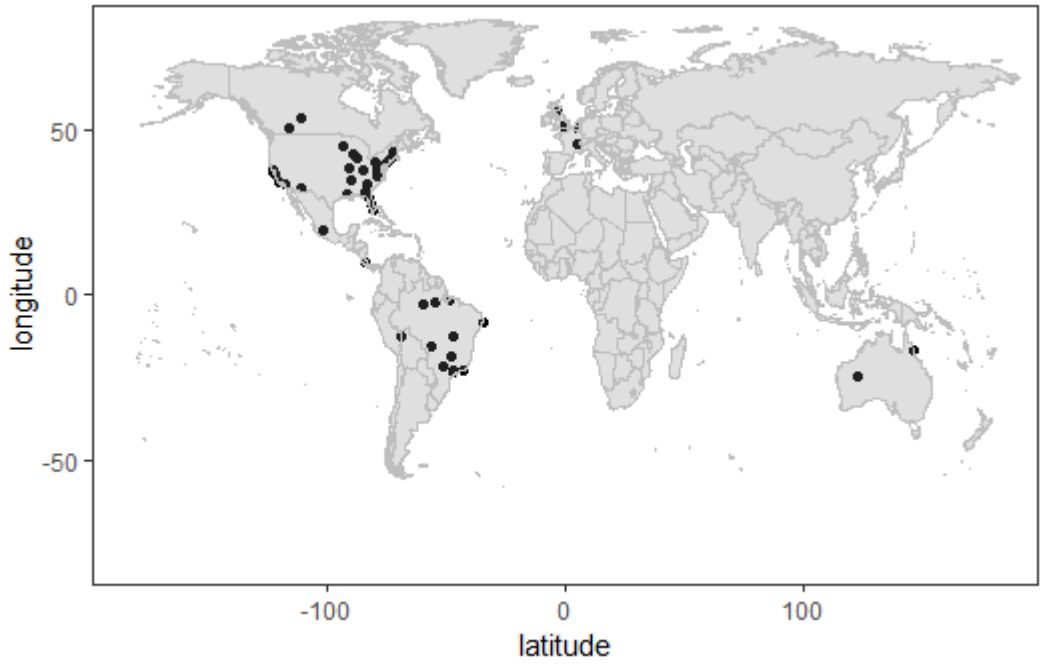
The function has one argument:

- **data**: This is the output from `authors_georef()`. Must be an object.

The output of `authors_georef()` is a ggplot object.

Example

```
plot_addresses_points <- plot_addresses_points(data, filename_root="./output/example")
```



2.4.3. Net Plots (base)

The `plot_net_coauthor()` function plots a co-authorship network based on the countries in which authors are based.

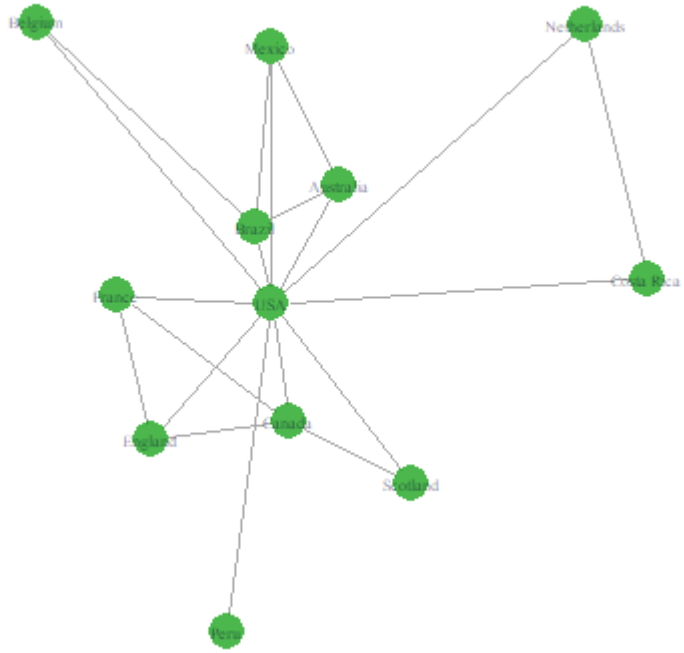
The function has one argument:

- **data**: This is the output from `authors_georef()`. Must be an object.

This function has one output, a plot, built in `igraph`.

Example

```
plot_addresses_points <- plot_addresses_points(data, filename_root="./output/example")
```



2.4.4. Net Plots (Country)

The `plot_net_country()` function plots a coauthorship network of the countries in which co-authors are based that is overlayed on a world map. The circles represent the number of authors based in a country.

The function has one argument:

- **data**: This is the output from `authors_georef()`. Must be an object.

The output of `plot_net_country()` is an list. The `$plot` element contains ggplot object. Because the ability to customize `$plot` is limited, three datasets are provided so that users can generate and customize their own plots:

1. The `$data_path` element contains the data for the connecting lines.
2. The `$data_polygon` element contains the data for the country outlines.
3. The `$data_point` element contains the data for the circles on the map.

Example

```
plot_net_country <- plot_net_country(data, filename_root="./output/example")
```

2.4.5. Net Plots (Addresses)

The `plot_net_addresses()` function is used to plot a georeferenced coauthorship network based on author institutional addresses. **Warning:** This function can create a large data set (100s of MB) and may takes several minutes to complete. . . be patient and take into account the system resources available when running it.

The function has one argument:

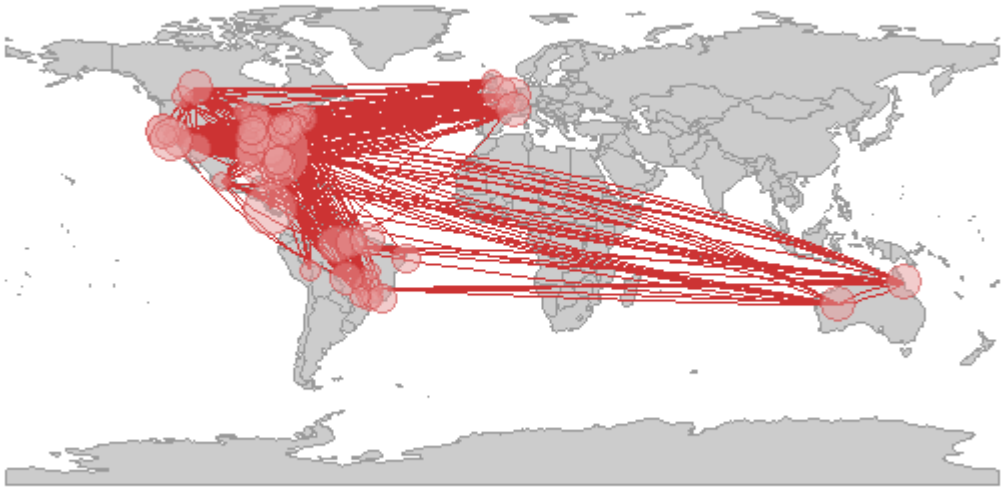
- **data**: This is the output from `authors_georef()`. Must be an object.

The output of `plot_net_addresses()` is an list. The `$plot` element contains ggplot object. Because the ability to customize `$plot` is limited, three datasets are provided so that users can generate and customize their own plots:

1. The `$data_path` element contains the data for the connecting lines.
2. The `$data_polygon` element contains the data for the country outlines.
3. The `$data_point` element contains the data for the circles on the map.

Example

```
plot_net_addresses <- plot_net_addresses(data, filename_root="./output/example")
```



Acknowledgments

Support for the development of refnet was provided by grants from the University of Florida Center for Latin American Studies and the University of Florida Informatics Institute.

References

Aria, M. & Cuccurullo, C. (2017) bibliometrix: An R-tool for comprehensive science mapping analysis. Journal of Informetrics, 11(4): 959-975.

Fortunato, S., C. T. Bergstrom, K. Barner, J. A. Evans, D. Helbing, S. Milojevic, A. M. Petersen, F. Radicchi, R. Sinatra, B. Uzzi, A. Vespignani, L. Waltman, D. Wang, & A.-L. Barabasi (2018). Science of science. Science, 359: eaao0185.

Lariviare, V., Ni, C., Gingras, Y., Cronin, B., & Sugimoto, C. R. (2013). Bibliometrics: Global gender disparities in science. Nature News, 504(7479): 211-213

R Core Team (2017). R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Smalheiser, N. R., & Torvik, V. I. (2009). Author name disambiguation. Annual Review of Information Science and Technology, 43(1): 1-43.

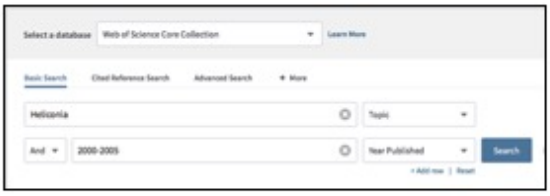
Smith, M. J., Weinberger, C., Bruna, E. M., & Allesina, S. (2014). The scientific impact of nations: Journal placement and citation performance, PLOS One 9(10): e109195.

Strotmann, A. and Zhao, D., (2012). Author name disambiguation: What difference does it make in author based citation analysis?. Journal of the Association for Information Science and Technology, 63(9): 1820-1833.

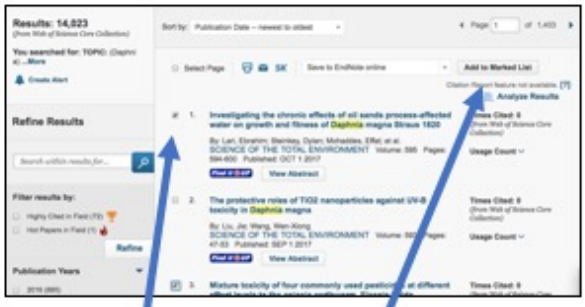
Sugimoto CR, Lariviare V. (2018). Measuring Research: What Everyone Needs to Know?. Oxford University Press, Oxford, UK. 149 pp.

Westgate, M. J. (2018b). revtools: bibliographic data visualization for evidence synthesis in R. bioRxiv:262881. doi: 10.1101/262881

Appendix 1: Guide to downloading reference records from the Web of Science.



(A) Search for articles of interest.



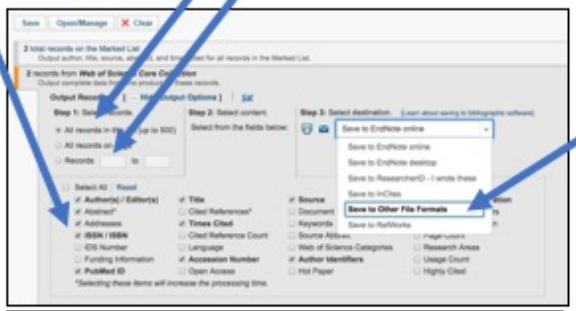
(B): Check the box by all publications of interest and add click this button to add them to the "Marked List".

(C) When finished searching, click on "Marked List" to prepare reference records for downloading.

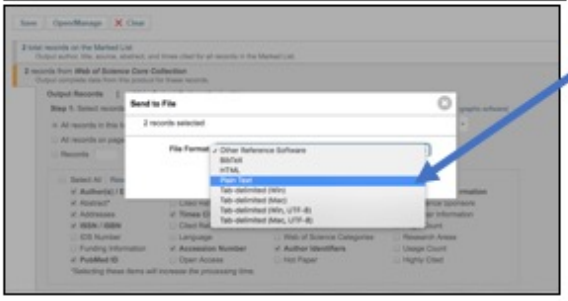
Warning: Do *not* download records using the dropdown menu on the search result page!

(D) Select "Addresses" on the "My List" page, along with any other data associated with reference records that are of interest.

(E) Up to 500 records can be downloaded from the WOS at a time.
Marked Lists of < 500 records: Select "All records in this list (up to 500)", then proceed to Steps F & G.
Marked Lists of > 500 records: Select 500 records by filling in the boxes next to "Records", complete Steps F & G, and repeat until all records are downloaded (e.g., "1" to "500" [F & G], "501" to 1000" [F & G],"1001" to "1500" [F & G],…. "3000" to "3261" [F & G]).



(F) Select "Save to Other File Formats" from the menu



(G) When the "Send to File" window appears, select "Plain Text" to save the file for that batch of records. If saving >500 records, be sure to change the name of each batch of records being downloaded.

Appendix 2: Web of Science Data Field Definitions

Table 1: Table 1. Definition of column headings in the output of read_references()[1]. Most are Web of Science Core Collection Field Tags associated with different data types.

Column Heading	Definition
filename	file from which records were imported
AB	Abstract
AF	Author Full Name
AU	Authors
BP	Beginning Page
C1	Author Address
CR	Cited References
DE	Author Keywords
DI	Digital Object Identifier (DOI)
EM	E-mail Address
EP	Ending Page
FN	File Name
FU	Funding Agency and Grant Number
PD	Publication Date
PG	Page Count
PT	Publication Type (J=Journal; B=Book; S=Series; P=Patent)
PU	Publisher
PY	Publication Year
RI	ResearcherID Number
OI	Open Researcher and Contributor ID Number (ORCID ID)
PM	PubMed ID
RP	Reprint Address
SC	Research Areas
SN	International Standard Serial Number (ISSN)
SO	Publication Name
TC	Web of Science Core Collection Times Cited Count
TI	Document Title
UT	Accession Number
VL	Volume
WC	Web of Science Categories
Z9	Total Times Cited Count[2]
refID	a unique identifier for each article in the dataset assigned by refnet

[1]the following Web of Science data fields are only included if users select the `all.fields=T` option in `references_read()`: CC, CH, CL, CT, CY, DT, FX, GA, GE, ID, IS, J9, JI, LA, LT, MC, MI, NR, PA, PI, PN, PS, RID, SU, TA, VR.

[2]Includes sitations in the Web of Science Core Collection, BIOSIS Citation Index, Chinese Science Citation Database, Data Citation Index, Russian Science Citation Index, and SciELO Citation Index.

Appendix 3: `authors_clean()` output

Table 2: Information provided by the `authors_clean()` function to help users assess the validity of groupings generated by `refnet`'s disambiguation algorithm.

Field	Defiition
authorID	AuthorID is a unique identifier for each name in the database (i.e., every
author of every	paper; the initial assumption of refnet's disambiguation
algorithm is tha	t all authors of all articles are different individuals).
AU	Authors
AF	Author Full Name
groupID	This indicates which names (i.e., AuthorID numbers) have been grouped
together under a	single groupID number because they are believed to be the
same person. Bec	ause disambiguation is performed iteratively. The lowest
authorID number	in a group will always be used as the groupID.
match_name	The name under which the algorithm groups all of an author's putative name
variants.	
similarity	NA
author_order	the location on the article's list of authors where this specific author is
found	
address	the author's complete address as listed in the record for an article.
university	the author's department, if one is listed in the address
department	the author's department, if one is listed in the address
short_address	the author's street address
postal_code	the author's postal code
country	the country in which an author's institution is based
RP_address	the reprint address, if present
RI	the author's Thomson-Reuters Researcher ID number in the WOS record for an
article(if they	have one).
OI	the author's ORCID ID number in the record for an article (if they have
one).	
EM	the author's email address in the WOS record for an article (if it lists
one).	
UT	Accession Number
refID	An id number given to each reference
PT	Publication Type (J=Journal; B=Book; S=Series; P=Patent)
PY	Publication Year
PU	Publisher

Appendix 4: Overview of the `refnet` author name disambiguation algorithm.

Name disambiguation is a complex process that is the subject of active research. There are a variety of approaches to disambiguation in large datasets; here we describe the alogorithm for parsing author adresses and disambiguating author names with the `authors_clean()` function.

There are three primary difficulties in assigning authors to their products in bibliometric databases like the Web of Science. First, not all authors have a unique identifier such as an ORCID iD (ORCID) or ResearcherID (RID). Second, an author's name can vary over time. It can also be reported inconsistently accross journals. For instance, while author last names are always reported, the first names might only be represented by initials and middle names (if authors have one) might not be reported or might only be stored as a middle initial. Finally, information in the "address" field can have inconsistent structures. In addition, only after 2008 did Web of Sceince records directly link each author with their institutional address. As a result, for pre-2008 Web of Science records it can be difficult to relate each author with their institutional address (the same is true for emaail addresses). In these cases, we have no way to reliably match addresses to authors using the information in the reference record and therefore insert 'Could not be extracted' in the address field. This does not mean an address was not listed or cannot be assigned after manual inspection - only that there was no way to discern to which author the address belongs. Coupled with

changes in author addresses as they change institutions, this the inconsistent or incomplete information associated with author addresses makes disambiguating names difficult.

To address this we've created a process to identify clusters or common authors by iteratively building webs of authors using any available information to link groups together. In this way we do not require an entry to contain all relevant fields, but can neverthtless create likely groupings and then refine them by throwing out obvious spurious connections. In the case of authors with ORCID and RID numbers, identifying commonalities is quite easy, and we hope that authors will contune to sign up for these identifiers to facilitate disambiguation.

The first step in our disambiguation process is matching all groups together with common ORCID and RID nunmbers. The remaining entries go through a series of logical rules to help match the author with only *likely* entries. Throughout this analysis we assume that every author has a complete last name and require the author's record contain any two of the following: first name, middle name, address, or email (the first and middle name can be initials). Requiring this type of information means we cannot match authors that do not contain any of this extra information, and so we do not group entries with no middle name, address, AND email, but instead call them their own group and skip them from the following analysis. To lower calculation times, as the algorithm attempts to match each entry it creates a subset of possible matching combinations of last and first names, and then attempts to match them against middle initials, address, and email addresses.

- *note regarding email addresses* - Similar to street addresses, email addresses are often stored inconsistently with no direct link between a specific author and specific email address. In these cases, we run a Jaro-Winkler distance measurement that calculates the amount of transpositions required to turn one string (an author name) into another (an email address). This works very well when email addresses are in a standard format (e.g., "lastname"."firstname" @ university.edu). We match author names to each email and use a threshold percentage of 0.70. If no names match up below this threshold we disregard the email and leave the field blank in the author name.

Below is an example of how the algorithm processes a sample data set.

	AF	groupID	address	OI	RI	email
1	Smith, J	1	100 University Rd, Austin, Tx			
2	Smith, Jon Karl	2		12345678		
3	Smith, Jon K	3			987654	smith.j@ut.edu
4	Smith, J. K	4				
5	Smith, J	2	100 University Rd, Austin, Tx	12345678		
6	Smith, J	6				
7	Smith, Jon	7				smith.j@ut.edu
8	Smith, James L.	8	300 Cross Street, New York, NY	11334578		
9	Smith, James	9				smither@smbc.org
10	Smith, Sam	10	100 University Rd, Austin, Tx			

In this dataset we have 10 authors, with a mixture of incomplete columns for each row. Rows 2 and 5 were given the same groupID priori because of their matching ORCID. We'll walk through the remaining entries and how the algorithm matches names together.

To lower the number of Type II errors we build a dataset of possible matches for each author; each entry in this subset must adhere to the following guidelines:

1. In all cases last names must match exactly (case insensitive). This means misspelled names will likely not get matched *unless* they have an ORCID or RID against which to match.
2. First names must match; in the case they only have an initial then that initial must match.
3. Middle names must match; in the case they only have an initial they must match. Cases of authors with no middle name are allowed if the author's record has another piece of identifying information (e.g., an address or email address).

Entry 1. In our test data we will start trying to match the first entry "Smith, J" in row 1. By subsetting with the above rules, we'd be matching the first row against rows 2, 3, 4, 5, 6, 7, 9:

1	Smith, J	1	100 University Rd, Austin, Tx			
	AF	groupID	address	OI	RI	email
2	Smith, Jon Karl	2		12345678		
3	Smith, Jon K	3			987654	smith.j@ut.edu
4	Smith, J. K	4				
5	Smith, J	2	100 University Rd, Austin, Tx	12345678		
6	Smith, J	6				
7	Smith, Jon	7				smith.j@ut.edu
9	Smith, James	9				smither@smbc.org

Once we have our subset of possible similar entries, we match the existing info of row 1 against the subset. The entry only needs to match one extra piece of information - either address, email, or middle name. If it matches we assume it is the same person, and change the groupID numbers to reflect this.

In our test data, there is only one piece of information we can match against - address, which makes the obvious match Row 5. We therefore change the groupID for our entry to groupID = 2. This gives us three entries with groupID = 2.

Entry 2. Row 2 was already matched to another group using ORCID prior, so it is skipped.

Entry 3. Row 3 has 2 unique identifying pieces of information: A middle initial and an email. This subset is smaller because we have a middle initial to filter out the Smith, J.L entries:

3	Smith, Jon K	3			987654	smith.j@ut.edu
	AF	groupID	address	OI	RI	email
1	Smith, J	2	100 University Rd, Austin, Tx			
2	Smith, Jon Karl	2		12345678		
4	Smith, J. K	4				
5	Smith, J	2	100 University Rd, Austin, Tx	12345678		
7	Smith, Jon	7				smith.j@ut.edu

Matching this information against our subset, the two possible matches are Row 2 and Row 7. In cases with multiple matches we choose the entry with the lowest number, as it is likely to have been grouped already. However, even if the entry in Row 7 was chosen as a match, it will eventually be matched up to groupID = 2. When a 'parent' groupID is changed in this way, all the 'child' entries are changed to match the new groupID as well. As such, the decision is partially arbitrary.

Entry 4 - This entry gets assigned groupID = 2 as well because it has a matching middle initial with Row 2 and Row 3:

4	Smith, J. K	4				
	AF	groupID	address	OI	RI	email
1	Smith, J	2	100 University Rd, Austin, Tx			
2	Smith, Jon Karl	2		12345678		
3	Smith, Jon K	2			987654	smith.j@ut.edu
5	Smith, J	2	100 University Rd, Austin, Tx	12345678		
6	Smith, J	6				
7	Smith, Jon	7				smith.j@ut.edu
8	Smith, James L.	8	300 Cross Street, New York, NY	11334578		
9	Smith, James	9				smither@smbc.org

Entry 5 - Row 5 has already been matched with ORCID, so it is skipped.

Entry 6 - Row 6 has no additional matching information - no middle name, address, or email. There is therefore no way to reliably know which 'Smith, J' it belongs to, so the entry is assumed to be its own unique group and is skipped.

Entry 7 - Entry 7 has one unique identifier: an email address. It gets matched to the entry in Row 3 and therefore is assigned groupID = 2.

7	Smith, Jon	7				smith.j@ut.edu
	AF	groupID	address	OI	RI	email
1	Smith, J	2	100 University Rd, Austin, Tx			
2	Smith, Jon Karl	2		12345678		
3	Smith, Jon K	2			987654	smith.j@ut.edu
4	Smith, J. K	2				
5	Smith, J	2	100 University Rd, Austin, Tx	12345678		
6	Smith, J	6				

After these first 7 entries, we've correctly matched all likely 'Smith, Jon Karl' together and created the 'Jon Karl Smith' complex. Now we'll move onto a situation where we have inadequate information, and must therefore run a Jaro-Winkler distance analysis to decide the likely match.

Entry 8 - This novel entry has two unique pieces of information: a middle initial and an ORCID. We know the ORCID did not match any previous entries, and the middle initial does not match up with any of the 'Smith' names in our record.

8	Smith, James L.	8	300 Cross Street, New York, NY	11334578		
	AF	groupID	address	OI	RI	email
1	Smith, J	1	100 University Rd, Austin, Tx			
5	Smith, J	2	100 University Rd, Austin, Tx	12345678		
6	Smith, J	6				
9	Smith, James	9				smither@smbc.org

Because there are no suitable matches using initial criteria, we instead match the entry by calculating a Jaro-Winkler

distance between the name in Row 8 and the author names in our subset. The results are: 0.9 [Row 1], 0.9 [Row 5], 0.9 [Row 6], and 0.96 [Row 9]. Therefore, the most likely match for the name 'Smith, James L' is the name in Row 9 ('Smith, James'). We change the groupID to 9, and set aside this entry for the user to manually review later.

Entry 9 - Results in the same result as 8 and is matched to entry 8 which already has the groupID of 9.

Entry 10 - This entry has no matching names and results in no change to the groupID number.

	AF	groupID	address	OI	RI	email
1	Smith, J	2	100 University Rd, Austin, Tx			
2	Smith, Jon Karl	2		12345678		
3	Smith, Jon K	2			987654	smith.j@ut.edu
4	Smith, J. K	2				
5	Smith, J	2	100 University Rd, Austin, Tx	12345678		
6	Smith, J	6				
7	Smith, Jon	2				smith.j@ut.edu
8	Smith, James L.	9	300 Cross Street, New York, NY	11334578		
9	Smith, James	9				smither@smbc.org
10	Smith, Sam	10	100 University Rd, Austin, Tx			

Thus our final results are: **[is something missing here?]**

As a final check against our created groupings, we attempt to prune groupings by reanalyzing if First name, last name, and middle names match. This is because matching entries using incomplete info occasionally creates novel situations where two similar names are called the same groupID if they have similar other information like same first name and the same address. Additionally, the imperfect matching of email addresses occasionally matches up relatives or significant others who publish together with the incorrect email creating a mismatched complex. This final pruning step separates the groups entirely by name. It should be noted this situation in general is rare.

This step is not necessary in our example data as all names in each grouping were logical. In our final data set, we have therefore identified 4 likely complexes (Smith, Jon Karl; Smith, J; Smith, James; and Smith Sam). Entrys 8 and 9 were matched together and require hand checking as no novel information was used. This results in this data.frame being outputted separate from the master authors data.frame:

	AF	groupID	address	OI	RI	email	similiarity	matchname
8	Smith, James L.	9	300 Cross Street, New York, NY	11334578			0.96	Smith, James
9	Smith, James	9				smither@smbc.org	0.96	Smith, James L.