SOFTWARE TOOL ARTICLE

epicontacts: Handling, visualisation and analysis of epidemiological contacts [version 2; referees: 1 approved, 1 approved with reservations]

VP Nagraj¹, Nistara Randhawa², Finlay Campbell³, Thomas Crellen⁴, Bertrand Sudre⁵, Thibaut Jombart³

¹Research Computing, University of Virginia School of Medicine, Charlottesville, VA, 22903, USA
²One Health Institute, University of California, Davis, Davis, CA, 95616, USA
³MRC Centre for Outbreak Analysis and Modelling, Department of Infectious Disease Epidemiology, School of Public Health, Imperial College London, London, W2 1PG, UK
⁴Mahidol-Oxford Tropical Medicine Research Unit, Bangkok , 10400, Thailand
⁵European Centre for Disease Prevention and Control, Stockholm, Sweden

Abstract
Epidemiological outbreak data is often captured in line list and contact format to facilitate contact tracing for outbreak control. epicontacts is an R package that provides a unique data structure for combining these data into a single object in order to facilitate more efficient visualisation and analysis. The package incorporates interactive visualisation functionality as well as network analysis techniques. Originally developed as part of the Hackout3 event, it is now developed, maintained and featured as part of the R Epidemics Consortium (RECON). The package is available for download from the Comprehensive R Archive Network (CRAN) and GitHub.

Keywords
contact tracing, outbreaks, R

This article is included in the RPackage gateway.

Open Peer Review

Referee Status: ? ✓

Invited Referees
1 2

version 2
published 11 Oct 2018

version 1
published 10 May 2018

Report

Report

1 Melissa A. Rolfes, Centers for Disease Control and Prevention (CDC), USA
2 Peter Adebayo Adewuyi, Liberia Field Epidemiology Training Program, Liberia African Field Epidemiology Network (AFENET), Uganda

Discuss this article

Comments (0)
Introduction

In order to study, prepare for, and intervene against disease outbreaks, infectious disease modellers and public health professionals need an extensive data analysis toolbox. Disease outbreak analytics involve a wide range of tasks that need to be linked together, from data collection and curation to exploratory analyses, and more advanced modelling techniques used for incidence forecasting\(^1\),\(^2\) or to predict the impact of specific interventions\(^3\),\(^4\). Recent outbreak responses suggest that for such analyses to be as informative as possible, they need to rely on a wealth of available data, including timing of symptoms, characterisation of key delay distributions (e.g. incubation period, serial interval), and data on contacts between patients\(^5\),\(^6\),\(^7\),\(^8\).

The latter type of data is particularly important for outbreak analysis, not only because contacts between patients are useful for unravelling the drivers of an epidemic\(^9\),\(^10\), but also because identifying new cases early can reduce ongoing transmission via contact tracing, i.e. follow-up of individuals who reported contacts with known cases\(^11\),\(^12\).

However, curating contact data and linking them to existing line lists of cases is often challenging, and tools for storing, handling, and visualising contact data are often missing\(^13\),\(^14\).

Here, we introduce epicontacts, an R\(^15\) package providing a suite of tools aimed at merging line lists and contact data, and providing basic functionality for handling, visualising and analysing epidemiological contact data. Maintained as part of the R Epidemics Consortium (RECON), the package is integrated into an ecosystem of tools for outbreak response using the R language.

Use cases

Those interested in using epicontacts should have a line list of cases as well as a record of contacts between individuals. Both datasets must be enumerated in tabular format with rows and columns. At minimum the line list requires one column with a unique identifier for every case. The contact list needs two columns for the source and destination of each pair of contacts. The datasets can include arbitrary features of case or contact beyond these columns. Once loaded into R and stored as data.frame objects, these datasets can be passed to the make_epicontacts() function (see ‘Methods’ section for more detail). For an example of data prepared in this format, users can refer to the outbreaks R package.

```r
# load the outbreaks package
library(outbreaks)

# example simulated ebola data
# line list
str(ebola_sim$linelist)

# contact list
str(ebola_sim$contacts)
```

Amendments from Version 1

In response to suggestions provided during the peer review process, the authors have made several updates to the manuscript. The body of the text is now organized more intuitively, introducing use cases for the epicontacts package before discussing specific functionality. Furthermore, the provenance of the data set is now described in the “Data handling” sub-section. The text has also been updated to include links to additional resources that demonstrate package usage. The authors feel that these changes have improved the manuscript and would like to thank the reviewers for providing their feedback.

See referee reports
The data handling, visualization, and analysis methods described above represent the bulk of epicontacts features. More examples of how the package can be used as well as demonstrations of additional features can be found through the RECON learn platform and the epicontacts vignettes.

### Methods

#### Operation

epicontacts is released as an open-source R package. A stable release is available for Windows, Mac and Linux operating systems via the CRAN repository. The latest development version of the package is available through the RECON Github organization. At minimum users must have R installed. No other system dependencies are required.

```r
# install from CRAN
install.packages("epicontacts")
# install from Github
install.packages("devtools")
devtools::install_github("reconhub/epicontacts")
```

#### Implementation

**Data handling.** epicontacts includes a novel data structure to accommodate line list and contact list datasets in a single object. This object is constructed with the `make_epicontacts()` function and includes attributes from the original datasets. Once combined, these are mapped internally in a graph paradigm as nodes and edges. The epicontacts data structure also includes a logical attribute for whether or not this resulting network is directed.

The package takes advantage of R’s generic functions, which call specific methods depending on the class of an object. This is implemented several places, including the `summary.epicontacts()` and `print.epicontacts()` methods, both of which are respectively called when the `summary()` or `print()` functions are used on an
The example that follows will use the `mers_korea_2015` dataset from `outbreaks`, which includes initial data collected by the Epidemic Intelligence group at European Centre for Disease Prevention and Control (ECDC) during the 2015 outbreak of Middle East respiratory syndrome (MERS-CoV) in South Korea. Note that the data used here was provided in `outbreaks` for teaching purposes, and therefore does not include the complete line list or contacts from the outbreak.

```r
# install the outbreaks package for data
globalInstall(pkgs="outbreaks")

# load the outbreaks package
library(outbreaks)

# construct an epicontacts object
x <- make_epicontacts(linelist=mers_korea_2015[[1]],
                      contacts = mers_korea_2015[[2]],
                      directed=TRUE)

# print the object
x
```

```
## // Epidemiological Contacts //
## // class: epicontacts
## // 162 cases in linelist; 98 contacts; directed
## // linelist
## # data.frame: 162 obs. of 15 variables:
## # $ id         : chr "SK_1" "SK_2" "SK_3" "SK_4" ...
## # $ age        : int 68 63 76 46 50 71 28 46 56 44 ...
## # $ age_class  : chr "60-69" "60-69" "70-79" "70-79" "40-49" ...
## # $ sex        : Factor w/ 2 levels "F","M": 2 1 2 1 2 1 2 2 2 2 ...
## # $ place_infect: Factor w/ 2 levels "Middle East",..: 1 2 2 2 2 2 2 2 2 2 ...
## # $ reporting_ctry: Factor w/ 2 levels "China","South Korea": 2 2 2 2 2 2 2 2 2 2 ...
## # $ loc_hosp    : Factor w/ 13 levels "365 Yeollin Clinic, Seoul",...
## # $ dt_onset    : Date, format: "2015-05-11" "2015-05-18" ...
## # $ dt_report   : Date, format: "2015-05-19" "2015-05-20" ...
## # $ week_report : Factor w/ 5 levels "2015_21","2015_22",...
## # $ dt_start_exp: Date, format: "2015-04-18" "2015-05-15" ...
## # $ dt_end_exp  : Date, format: "2015-05-04" "2015-05-20" ...
## # $ dt_diag     : Date, format: "2015-05-20" "2015-05-20" ...
## # $ outcome     : Factor w/ 2 levels "Alive","Dead": 1 1 2 1 2 1 1 1 1 ...
## # $ dt_death    : Date, format: NA NA ...
## // contacts
## # data.frame: 98 obs. of 4 variables:
## # $ from       : chr "SK_14" "SK_14" "SK_14" "SK_14" ...
## # $ to         : chr "SK_113" "SK_116" "SK_41" "SK_112" ...
## # $ exposure   : Factor w/ 5 levels "Contact with HCW",...
## # $ diff_dt_onset: int 10 13 14 14 15 15 16 16 16 ...

# view a summary of the object
summary(x)
```

```
## // Overview //
## // number of unique IDs in linelist: 162
## // number of unique IDs in contacts: 97
## // number of unique IDs in both: 97
## // number of contacts: 98
## // contacts with both cases in linelist: 100 %
```
Data visualisation. epicontacts implements two interactive network visualisation packages: visNetwork and threejs\textsuperscript{TM}. These frameworks provide R interfaces to the vis.js and three.js JavaScript libraries respectively. Their functionality is incorporated in the generic plot() method (Figure 1) for an epicontacts object, which can be toggled between either with the “type” parameter. Alternatively, the visNetwork interactivity is accessible via vis_epicontacts() (Figure 2), and threejs through graph3D() (Figure 3). Each function has a series of arguments that can also be passed through plot(). Both share a color palette, and users can specify node, edge and background colors. However, vis_epicontacts()
Figure 2. The vis_epicontacts() function explicitly calls visNetwork to make an interactive plot of the contact network.

Figure 3. The graph3D() function generates a three-dimensional network plot.
includes a specification for “node_shape” by a line list attribute as well as a customization of that shape with an icon from the Font Awesome icon library. The principal distinction between the two is that `graph3D()` is a three-dimensional visualisation, allowing users to rotate clusters of nodes to better inspect their relationships.

```r
plot(x)

vis_epicontacts(x,
    node_shape = "sex",
    shapes = c(F = "female", M = "male"),
    edge_label = "exposure")

graph3D(x, bg_col = "black")
```

**Data analysis.** Subsetting is a typical preliminary step in data analysis. `epicontacts` leverages a customized `subset` method to filter line lists or contacts based on values of particular attributes from nodes, edges or both. If users are interested in returning only contacts that appear in the line list (or vice versa), the `thin()` function implements such logic.

```r
# subset for males
subset(x, node_attribute = list("sex" = "M"))

# subset for exposure in emergency room
subset(x, edge_attribute = list("exposure" = "Emergency room"))

# subset for males who survived and were exposed in emergency room
subset(x,
    node_attribute = list("sex" = "M", "outcome" = "Alive"),
    edge_attribute = list("exposure" = "Emergency room"))

thin(x, "contacts")
thin(x, "linelist")
```

For analysis of pairwise contact between individuals, the `get_pairwise()` feature searches the line list based on the specified attribute. If the given column is a numeric or date object, the function will return a vector containing the difference of the values of the corresponding “from” and “to” contacts. This can be particularly useful, for example, if the line list includes the date of onset of each case. The subtracted value of the contacts would approximate the serial interval for the outbreak. For factors, character vectors and other non-numeric attributes, the default behavior is to print the associated line list attribute for each pair of contacts. The function includes a further parameter to pass an arbitrary function to process the specified attributes. In the case of a character vector, this can be helpful for tabulating information about different contact pairings with `table()`.

```r
# find interval between date onset in cases
get_pairwise(x, "dt_onset")

# find pairs of age category contacts
get_pairwise(x, "age_class")

# tabulate the pairs of age category contacts
get_pairwise(x, "age_class", f = table)
```

**Discussion**

**Benefits**

While there are software packages available for epidemiological contact visualisation and analysis, none aim to accommodate line list and contact data as purposively as `epicontacts`. Furthermore, this package strives to solve a problem of plotting dense graphs by implementing interactive network visualisation tools. A static plot of a network with many nodes and edges may be difficult to interpret. However, by rotating or hovering over an `epicontacts` visualisation, a user may better understand the data.

**Future considerations**

The maintainers of `epicontacts` anticipate new features and functionality. Future development could involve performance optimization for visualising large networks, as generating these interactive plots is resource intensive. Additionally, attention may be directed towards inclusion of alternative visualisation methods.

**Conclusions**

`epicontacts` provides a unified interface for processing, visualising and analyzing disease outbreak data in the R language. The package and its source are freely available on CRAN and GitHub. By developing functionality
with line list and contact list data in mind, the authors aim to enable more efficient epidemiological outbreak analyses.

Software availability
Software available from: https://CRAN.R-project.org/package=epicontacts
Source code available from: https://github.com/reconhub/epicontacts
Archived source code as at time of publication: https://zenodo.org/record/1210993
Software license: GPL 2

Grant information
The author(s) declared that no grants were involved in supporting this work.

Acknowledgements
The authors would like to thank all of the organizers and participants of the Hackout3 event held in Berkeley, California June 20–24, 2016. In particular, the authors acknowledge the support of the following organizations: MRC Centre for Outbreak Analysis, and Modelling at Imperial College London, the NIHR’s Modelling Methodology Health Protection Research Unit at Imperial College London, and the Berkeley Institute for Data Science.

References

21. Carroll LN, Au AP, Detwiler LT, et al.: Visualization and analytics

Page 9 of 13
PubMed Abstract | Publisher Full Text | Free Full Text

PubMed Abstract | Publisher Full Text

Peter Adebayo Adewuyi 1,2
1 Liberia Field Epidemiology Training Program, Monrovia, Liberia
2 African Field Epidemiology Network (AFENET), Kampala, Uganda

This is a good software developed which could help in continuous visualization of contacts and their progression in disease tracking.

It is user friendly for those who are not computer specialist and still want to visualize data.

Data visualization is pertinent to disease monitoring and what the authors have done will aid in helping epidemiologist and public health specialist involved in outbreak response to quickly visualize progression and spread of disease from primary to secondary contacts and how the disease is evolving among contacts.

The software will actually achieve its purpose as stated in the conclusion of the write-up. Good work done.

Is the rationale for developing the new software tool clearly explained?
Yes

Is the description of the software tool technically sound?
Yes

Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Yes

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Yes

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

**Competing Interests:** No competing interests were disclosed.
I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard.

Referee Report 31 May 2018

doi:10.5256/f1000research.15777.r34084

Melissa A. Rolfes
Centers for Disease Control and Prevention (CDC), Atlanta, GA, USA

The article describes an R-based software tool aimed to facilitate analysis of data from outbreaks that include line lists of cases and case-contact data. The R package, epicontacts, is part of a larger suite of tools housed at the R Epidemics Consortium (RECON). The epicontacts package has the ability to merge data about cases in a line list with case-contact details, which then allows the user to describe and visualize contact networks, incubation periods, and serial intervals within an outbreak.

The codes and methods for analysis are partly described in the article, and the authors should provide a link to the packages documentation, either at CRAN or RECON webpages, where readers could learn more about the package and its options.

The output of the package provided in the article was interesting and intriguing. I felt that it was only partly explained and the article could benefit from the authors annotating the output and its interpretation a bit further. I have explored the RECON website and found the RECON Learn modules to be quite helpful in providing annotation of the epicontacts output and some guidance on interpretation. I would recommend that the authors consider either expanding the annotation of the output in this article or explicitly direct readers to the RECON Learn website for further instruction.

Additional suggestions:
• Consider moving the section of the article called "Use cases" to before the "Data handling" subsection of the "Implementation" section. I felt that the description of the input datasets under "Use cases" was very informative and would have been organizational more helpful had it been placed earlier in the article.

• Consider describing the sample outbreak data in a bit further detail. It appears to be data describing the MERS outbreak that occurred in South Korea in 2015. I think the description should include whether the data are simulated or from a real outbreak (if from a real outbreak, then a reference to the outbreak description should be included), the scenario of the outbreak, how many cases, how many contacts, place of the outbreak, duration of the outbreak, and a brief description of the demographic details included in the dataset. This amount of detail would allow the reader to translate the details of the outbreak from your text to the output provided by epicontacts.

Is the rationale for developing the new software tool clearly explained? Yes

Is the description of the software tool technically sound? Yes
Are sufficient details of the code, methods and analysis (if applicable) provided to allow replication of the software development and its use by others?
Partly

Is sufficient information provided to allow interpretation of the expected output datasets and any results generated using the tool?
Partly

Are the conclusions about the tool and its performance adequately supported by the findings presented in the article?
Yes

**Competing Interests:** No competing interests were disclosed.

I have read this submission. I believe that I have an appropriate level of expertise to confirm that it is of an acceptable scientific standard, however I have significant reservations, as outlined above.

---

The benefits of publishing with F1000Research:

- Your article is published within days, with no editorial bias
- You can publish traditional articles, null/negative results, case reports, data notes and more
- The peer review process is transparent and collaborative
- Your article is indexed in PubMed after passing peer review
- Dedicated customer support at every stage

For pre-submission enquiries, contact research@f1000.com