epicontacts: Handling, visualisation and analysis of epidemiological contacts

VP Nagraj[1], Nistara Randhawa[2], Finlay Campbell[3], Thomas Crellen[4], Bertrand Sudre[5], and Thibaut Jombart[3]

[1]**School of Medicine Research Computing, University of Virginia, USA**
[2]**One Health Institute, University of California, Davis, USA**
[3]**MRC Centre for Outbreak Analysis and Modelling, Department of Infectious Disease Epidemiology, School of Public Health, Imperial College London, United Kingdom.**
[4]**Mahidol-Oxford Tropical Medicine Research Unit, Bangkok, Thailand**
[5]**European 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

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 `http://www.repidemicsconsortium.org/`), 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.

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

# example simulated ebola data

# line list
str(ebola_sim$linelist)
```

```
## 'data.frame':    5888 obs. of  9 variables:
##  $ case_id                : chr  "d1fafd" "53371b" "f5c3d8" "6c286a" ...
##  $ generation             : int  0 1 1 2 2 0 3 3 2 3 ...
##  $ date_of_infection      : Date, format: NA "2014-04-09" ...
##  $ date_of_onset          : Date, format: "2014-04-07" "2014-04-15" ...
##  $ date_of_hospitalisation: Date, format: "2014-04-17" "2014-04-20" ...
##  $ date_of_outcome        : Date, format: "2014-04-19" NA ...
##  $ outcome                : Factor w/ 2 levels "Death","Recover": NA NA 2 1 2 NA 2 1 2 1 ...
##  $ gender                 : Factor w/ 2 levels "f","m": 1 2 1 1 1 1 1 1 2 2 ...
##  $ hospital               : Factor w/ 11 levels "Connaught Hopital",..: 4 2 7 NA 7 NA 2 9 7 11 ...
```

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

```
## 'data.frame':    3800 obs. of  3 variables:
##  $ infector: chr  "d1fafd" "cac51e" "f5c3d8" "0f58c4" ...
##  $ case_id : chr  "53371b" "f5c3d8" "0f58c4" "881bd4" ...
##  $ source  : Factor w/ 2 levels "funeral","other": 2 1 2 2 2 1 2 2 2 2 ...
```

```
# example middle east respiratory syndrome data

# line list
str(mers_korea_2015$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" "40-49" ...
##  $ sex           : Factor w/ 2 levels "F","M": 2 1 2 1 2 2 1 1 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 1 ...
##  $ loc_hosp      : Factor w/ 13 levels "365 Yeollin Clinic, Seoul",..: 10 10 10 10 1 10 10 13 10
##  $ 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",..: 1 1 1 2 2 2 2 2 2 ...
##  $ 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 1 2 1 1 1 1 ...
##  $ dt_death      : Date, format: NA NA ...
```

```
# contact list
str(mers_korea_2015$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",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ diff_dt_onset: int  10 13 14 14 15 15 15 16 16 16 ...
```

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.

```
# install from CRAN
install.packages("epicontacts")

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

```
# load and attach the package
library(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_epiconctacts()` 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 `epicontacts` object. The package does not include built-in data, as exemplary contact and line list datasets are available in the `outbreaks` package[16].

```
# install the outbreaks package for data
install.packages("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" "40-49" ...
##  $ sex            : Factor w/ 2 levels "F","M": 2 1 2 1 2 2 1 1 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 1 ...
##  $ loc_hosp       : Factor w/ 13 levels "365 Yeollin Clinic, Seoul",..: 10 10 10 10 1 10 10 13 10
##  $ 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",..: 1 1 1 2 2 2 2 2 2 2 ...
##  $ 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 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",..: 2 2 2 2 2 2 2 2 2 2 ...
##  $ diff_dt_onset: int  10 13 14 14 15 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 %
##
## /// Degrees of the network //
##   // in-degree summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    1.00    1.00    1.01    1.00    3.00
##
##   // out-degree summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00    1.01    0.00   38.00
##
```

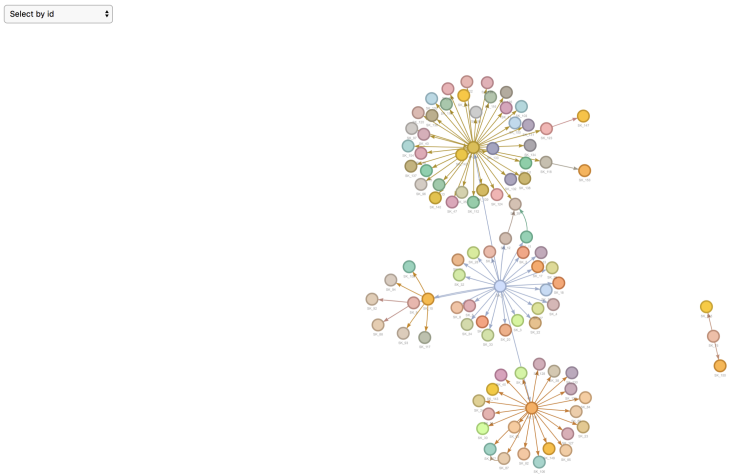

Figure 1. The generic plot() method for an epicontacts object will use the visNetwork method by default.

```
##   // in and out degree summary:
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   1.000   1.000   2.021   1.000  39.000
##
## /// Attributes //
##   // attributes in linelist:
##  age age_class sex place_infect reporting_ctry loc_hosp dt_onset dt_report week_report dt_start_e
##
##   // attributes in contacts:
##  exposure diff_dt_onset
```

Data visualisation

`epicontacts` implements two interactive network visualisation packages: `visNetwork` and `threejs` [17,18]. 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()` 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.

```
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.

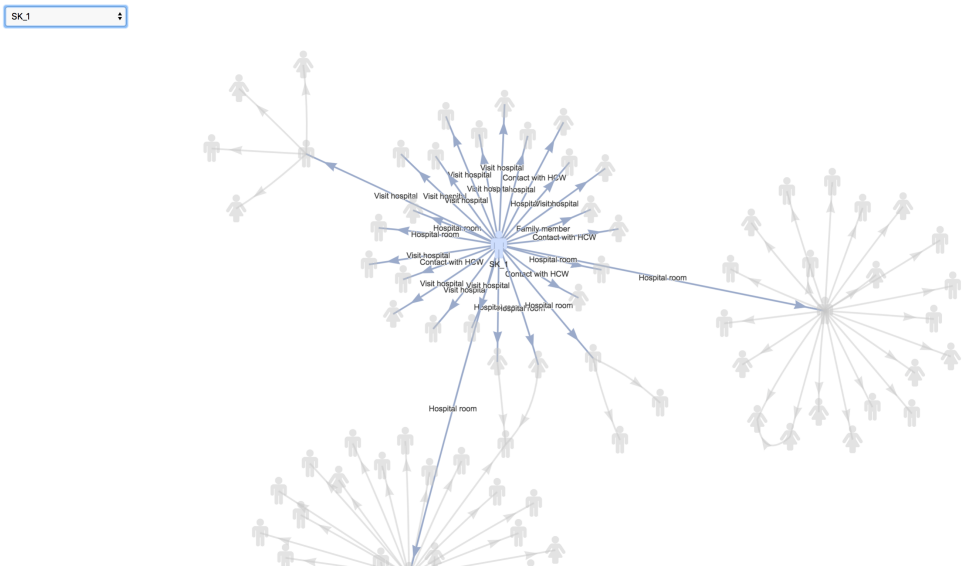

Figure 2. The vis_epicontacts() function explicitly calls visNetwork to make an interactive plot of the contact network.

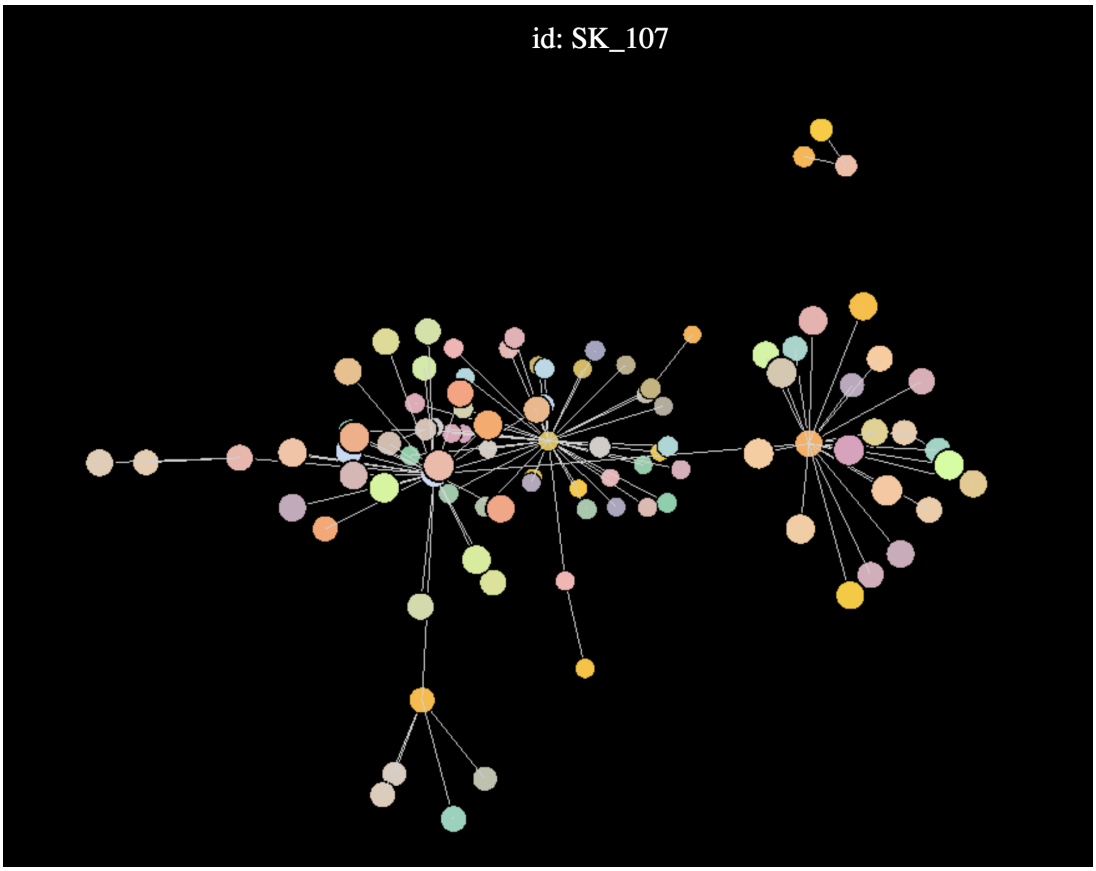

Figure 3. The graph3D() function generates a three-dimensional network plot.

```
# 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[19]. 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()`.

```
# 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`[20,21,22]. 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

Author contributions

- VPN: Conceptualization, Software, Writing - Original Draft Preparation
- NR: Conceptualization, Software, Writing - Original Draft Preparation
- FC: Conceptualization, Software, Writing - Original Draft Preparation
- TC: Conceptualization, Software
- BS: Conceptualization
- TJ: Conceptualization, Software, Writing - Original Draft Preparation

Competing interests

No competing interests were disclosed.

Grant information

The authors 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

[1] S. Funk, A. Camacho, A. J. Kucharski, R. M. Eggo, and W. J. Edmunds. Real-time forecasting of infectious disease dynamics with a stochastic semi-mechanistic model. *Epidemics*, Dec 2016.

[2] P. Nouvellet, A. Cori, T. Garske, I. M. Blake, I. Dorigatti, W. Hinsley, T. Jombart, H. L. Mills, G. Nedjati-Gilani, M. D. Van Kerkhove, C. Fraser, C. A. Donnelly, N. M. Ferguson, and S. Riley. A simple approach to measure transmissibility and forecast incidence. *Epidemics*, Feb 2017.

[3] P. Nouvellet, T. Garske, H. L. Mills, G. Nedjati-Gilani, W. Hinsley, I. M. Blake, M. D. Van Kerkhove, A. Cori, I. Dorigatti, T. Jombart, S. Riley, C. Fraser, C. A. Donnelly, and N. M. Ferguson. The role of rapid diagnostics in managing Ebola epidemics. *Nature*, 528(7580):S109–116, Dec 2015.

[4] E. P. Parker, N. A. Molodecky, M. Pons-Salort, K. M. O'Reilly, and N. C. Grassly. Impact of inactivated poliovirus vaccine on mucosal immunity: implications for the polio eradication endgame. *Expert Rev Vaccines*, 14(8):1113–1123, 2015.

[5] S. Cauchemez, C. Fraser, M. D. Van Kerkhove, C. A. Donnelly, S. Riley, A. Rambaut, V. Enouf, S. van der Werf, and N. M. Ferguson. Middle East respiratory syndrome coronavirus: quantification of the extent of the epidemic, surveillance biases, and transmissibility. *Lancet Infect Dis*, 14(1):50–56, Jan 2014.

[6] B. Aylward, P. Barboza, L. Bawo, E. Bertherat, P. Bilivogui, I. Blake, R. Brennan, S. Briand, J. M. Chakauya, K. Chitala, R. M. Conteh, A. Cori, A. Croisier, J. M. Dangou, B. Diallo, C. A. Donnelly, C. Dye, T. Eckmanns, N. M. Ferguson, P. Formenty, C. Fuhrer, K. Fukuda, T. Garske, A. Gasasira, S. Gbanyan, P. Graaff, E. Heleze, A. Jambai, T. Jombart, F. Kasolo, A. M. Kadiobo, S. Keita, D. Kertesz, M. Kone, C. Lane, J. Markoff, M. Massaquoi, H. Mills, J. M. Mulba, E. Musa, J. Myhre, A. Nasidi, E. Nilles, P. Nouvellet, D. Nshimirimana, I. Nuttall, T. Nyenswah, O. Olu, S. Pendergast, W. Perea, J. Polonsky, S. Riley, O. Ronveaux, K. Sakoba, R. Santhana Gopala Krishnan, M. Senga, F. Shuaib, M. D. Van Kerkhove, R. Vaz, N. Wijekoon Kannangarage, and Z. Yoti. Ebola virus disease in West Africa–the first 9 months of the epidemic and forward projections. *N. Engl. J. Med.*, 371(16):1481–1495, 10 2014.

[7] J. Agua-Agum, A. Ariyarajah, B. Aylward, I. M. Blake, R. Brennan, A. Cori, C. A. Donnelly, I. Dorigatti, C. Dye, T. Eckmanns, N. M. Ferguson, P. Formenty, C. Fraser, E. Garcia, T. Garske, W. Hinsley, D. Holmes, S. Hugonnet, S. Iyengar, T. Jombart, R. Krishnan, S. Meijers, H. L. Mills, Y. Mohamed, G. Nedjati-Gilani, E. Newton, P. Nouvellet, L. Pelletier, D. Perkins, S. Riley, M. Sagrado, J. Schnitzler, D. Schumacher, A. Shah, M. D. Van Kerkhove, O. Varsaneux, and N. Wijekoon Kannangarage. West African Ebola epidemic after one year–slowing but not yet under control. *N. Engl. J. Med.*, 372(6):584–587, Feb 2015.

[8] A. Cori, C. A. Donnelly, I. Dorigatti, N. M. Ferguson, C. Fraser, T. Garske, T. Jombart, G. Nedjati-Gilani, P. Nouvellet, S. Riley, M. D. Van Kerkhove, H. L. Mills, and I. M. Blake. Key data for outbreak evaluation: building on the Ebola experience. *Philos. Trans. R. Soc. Lond., B, Biol. Sci.*, 372(1721), May 2017.

[9] J. Agua-Agum, A. Ariyarajah, B. Aylward, L. Bawo, P. Bilivogui, I. M. Blake, R. J. Brennan, A. Cawthorne, E. Cleary, P. Clement, R. Conteh, A. Cori, F. Dafae, B. Dahl, J. M. Dangou, B. Diallo, C. A. Donnelly, I. Dorigatti, C. Dye, T. Eckmanns, M. Fallah, N. M. Ferguson, L. Fiebig, C. Fraser, T. Garske, L. Gonzalez, E. Hamblion, N. Hamid, S. Hersey, W. Hinsley, A. Jambei, T. Jombart, D. Kargbo, S. Keita, M. Kinzer, F. K. George, B. Godefroy, G. Gutierrez, N. Kannangarage, H. L. Mills, T. Moller, S. Meijers, Y. Mohamed, O. Morgan, G. Nedjati-Gilani, E. Newton, P. Nouvellet, T. Nyenswah, W. Perea, D. Perkins, S. Riley, G. Rodier, M. Rondy, M. Sagrado, C. Savulescu, I. J. Schafer, D. Schumacher, T. Seyler, A. Shah, M. D. Van Kerkhove, C. S. Wesseh, and Z. Yoti. Exposure Patterns Driving Ebola Transmission in West Africa: A Retrospective Observational Study. *PLoS Med.*, 13(11):e1002170, Nov 2016.

[10] S. Cauchemez, P. Nouvellet, A. Cori, T. Jombart, T. Garske, H. Clapham, S. Moore, H. L. Mills, H. Salje, C. Collins, I. Rodriquez-Barraquer, S. Riley, S. Truelove, H. Algarni, R. Alhakeem, K. AlHarbi, A. Turkistani, R. J. Aguas, D. A. Cummings, M. D. Van Kerkhove, C. A. Donnelly, J. Lessler, C. Fraser, A. Al-Barrak, and N. M. Ferguson. Unraveling the drivers of MERS-CoV transmission. *Proc. Natl. Acad. Sci. U.S.A.*, 113(32):9081–9086, 08 2016.

[11] M. Senga, A. Koi, L. Moses, N. Wauquier, P. Barboza, M. D. Fernandez-Garcia, E. Engedashet, F. Kuti-George, A. D. Mitiku, M. Vandi, D. Kargbo, P. Formenty, S. Hugonnet, E. Bertherat, and C. Lane. Contact tracing performance during the Ebola virus disease outbreak in Kenema district, Sierra Leone. *Philos. Trans. R. Soc. Lond., B, Biol. Sci.*, 372(1721), May 2017.

[12] S. Saurabh and S. Prateek. Role of contact tracing in containing the 2014 Ebola outbreak: a review. *Afr Health Sci*, 17(1):225–236, Mar 2017.

[13] World Health Organization. Response to measles outbreaks in measles mortality reduction settings: Immunization, vaccines and biologicals, 2009.

[14] P. Rakesh, D. Sherin, H. Sankar, M. Shaji, S. Subhagan, and S. Salila. Investigating a community-wide outbreak of hepatitis a in India. *J Glob Infect Dis*, 6(2):59–64, Apr 2014.

[15] R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2017. URL `https://www.R-project.org/`.

[16] Thibaut Jombart, Simon Frost, Pierre Nouvellet, Finlay Campbell, and Bertrand Sudre. *outbreaks: A Collection of Disease Outbreak Data*, 2017. URL `https://CRAN.R-project.org/package=outbreaks`. R package version 1.3.0.

[17] Almende B.V., Benoit Thieurmel, and Titouan Robert. *visNetwork: Network Visualization using 'vis.js' Library*, 2018. URL `https://CRAN.R-project.org/package=visNetwork`. R package version 2.0.3.

[18] B. W. Lewis. *threejs: Interactive 3D Scatter Plots, Networks and Globes*, 2017. URL `https://CRAN.R-project.org/package=threejs`. R package version 0.3.1.

[19] P. E. Fine. The interval between successive cases of an infectious disease. *Am. J. Epidemiol.*, 158(11):1039–1047, Dec 2003.

[20] M. Noremark and S. Widgren. EpiContactTrace: an R-package for contact tracing during livestock disease outbreaks and for risk-based surveillance. *BMC Vet. Res.*, 10:71, Mar 2014.

[21] L. N. Carroll, A. P. Au, L. T. Detwiler, T. C. Fu, I. S. Painter, and N. F. Abernethy. Visualization and analytics tools for infectious disease epidemiology: a systematic review. *J Biomed Inform*, 51:287–298, Oct 2014.

[22] J. L. Guthrie, D. C. Alexander, A. Marchand-Austin, K. Lam, M. Whelan, B. Lee, C. Furness, E. Rea, R. Stuart, J. Lechner, M. Varia, J. McLean, and F. B. Jamieson. Technology and tuberculosis control: the OUT-TB Web experience. *J Am Med Inform Assoc*, 24(e1):e136–e142, Apr 2017.