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. `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 visualization and analysis. The package incorporates interactive visualization 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 as well as public health professionals need an extensive toolbox for analyzing epidemiological data. Disease outbreak analytics involve a wide range of tasks which 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 in outbreak context, not only because contacts between patients is useful for unravelling the drivers of an epidemic[9,10], but also as a mean to identify new cases early and 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 linelists of cases is often challenging, and analytics tools for storing, handling, and visualising contact data are often missing[13,14].

Here, we introduce epicontacts, an R package providing a suite of tools aimed at merging linelists and contact data, and providing basic functionalities for handling, visualizing and analysing epidemiological contact data. Epicontacts is developed as part of the R Epidemics Consortium (RECON: `http://www.repidemicsconsortium.org/`), and as such is integrated alongside a larger set of analytics tools for outbreak response using the R software[15].

Methods

Operation

The `epicontacts` package is released as an open-source R package. A stable release is freely available for install on Windows, Mac and Linux operating systems via the CRAN repository. The latest development version of the package is available as part of the the RECON Github organization. From within R, users can issue the following commands to install the CRAN or Github version respectively:

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

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

Once installed, the package is ready to be loaded and attached. It includes vignettes describing use-cases and documentation for specific functions.

```
# load and attach the package
library(epicontacts)
```

```
# view vignettes
browseVignettes(package = "epicontacts")

# access function documentation
?make_epicontacts
?vis_epicontacts
```

Implementation

Data handling

The package is designed to handle contact tracing data that is organized in linelist and contact list format. As such, it includes a novel data structure to accommodate both of these datasets in a single object. `epicontacts` was designed and exported as an S3 class using an object oriented programing (OOP) approach in R. Based on the language's `list` data type, objects of this class are constructed with the `make_epiconctacts()` function and include attributes for line list (`data.frame`) and contact list (`data.frame`). Once combined in a single object, these are mapped via other functions 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.

With the line list and contact data in a single object, the `epicontacts` package takes advantage of R's implementation of generic functions, which call specific methods depending on the S3 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 deliberately does not include built-in contact and line list datasets, as these are abstracted in the `outbreaks` package[16]. The example that follows demonstrates how to do preliminary handling of data from that package with `epicontacts`.

```
# 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
##
## # A tibble: 162 x 15
##    id      age age_class sex   place_infect  reporting_ctry loc_hosp
##  * <chr> <int> <chr>     <fct> <fct>         <fct>          <fct>
##  1 SK_1     68 60-69     M     Middle East   South Korea    Pyeongtaek St~
##  2 SK_2     63 60-69     F     Outside Midd~ South Korea    Pyeongtaek St~
##  3 SK_3     76 70-79     M     Outside Midd~ South Korea    Pyeongtaek St~
##  4 SK_4     46 40-49     F     Outside Midd~ South Korea    Pyeongtaek St~
##  5 SK_5     50 50-59     M     Outside Midd~ South Korea    365 Yeollin C~
##  6 SK_6     71 70-79     M     Outside Midd~ South Korea    Pyeongtaek St~
##  7 SK_7     28 20-29     F     Outside Midd~ South Korea    Pyeongtaek St~
##  8 SK_8     46 40-49     F     Outside Midd~ South Korea    Seoul Clinic,~
##  9 SK_9     56 50-59     M     Outside Midd~ South Korea    Pyeongtaek St~
## 10 SK_10    44 40-49     M     Outside Midd~ China          Pyeongtaek St~
## # ... with 152 more rows, and 8 more variables: dt_onset <date>, dt_report
## #   <date>, week_report <fct>, dt_start_exp <date>, dt_end_exp <date>,
## #   dt_diag <date>, outcome <fct>, dt_death <date>
##
##   // contacts
##
## # A tibble: 98 x 4
##    from  to     exposure       diff_dt_onset
##    <chr> <chr>  <fct>                  <int>
##  1 SK_14 SK_113 Emergency room            10
##  2 SK_14 SK_116 Emergency room            13
##  3 SK_14 SK_41  Emergency room            14
##  4 SK_14 SK_112 Emergency room            14
##  5 SK_14 SK_100 Emergency room            15
##  6 SK_14 SK_114 Emergency room            15
##  7 SK_14 SK_136 Emergency room            15
##  8 SK_14 SK_47  Emergency room            16
##  9 SK_14 SK_110 Emergency room            16
## 10 SK_14 SK_122 Emergency room            16
## # ... with 88 more rows
```

```
# 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
##
##   // 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

As mentioned previously, the structure of the `epicontacts` object lends itself to network visualization. The package implements two interactive graph plotting packages: `visNetwork` and `threejs`[17,18]. These are htmlwidgets that provide R interfaces to JavaScript libraries, `vis.js` and `three.js` respectively. Their functionality is incorporated in the generic `plot()` method (see Figure 1) for an `epicontacts` object, which can be toggled between either with the "type" parameter. Alternatively, the `visNetwork` interactivity is accessible by using `vis_epicontacts()` (see Figure 2), and `threejs` via `graph3D()` (see Figure 3). Each function has a series arguments that can also be passed through `plot()`. Both share a color palette, and users can specify node, edge and background colors. However, these functions do have subtle differentiations in behaviors. For instance, `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 visualization. In addition to zooming and dragging the network, users can 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

One typical step for analyzing a dataset is to look at a certain subset of the data. `epicontacts` includes implementations of the generic R `subset` function to filter the line list or contacts based on values of particular attributes. These specifications are passed as named lists to the "node_attribute" and "edge_attribute" arguments, which can be used simultaneously if necessary. In addition to subsetting based on specific values, users may be interested in returning only contacts that appear in the line list or vice versa. The `thin()` function implements such logic.

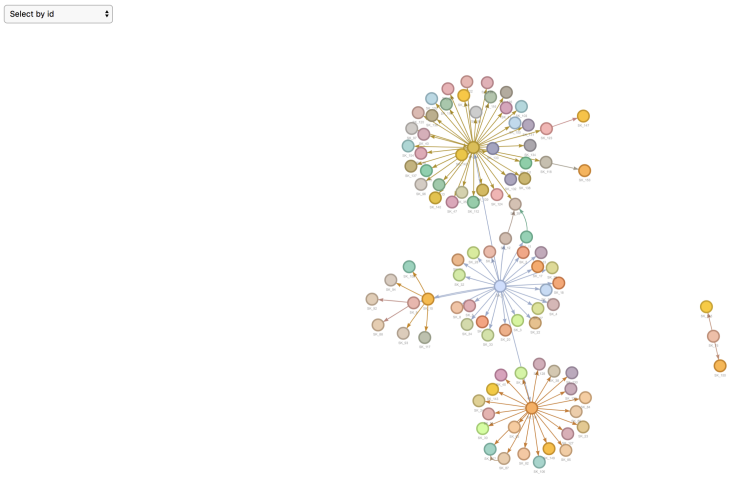

Figure 1. The generic plot() method for an epicontacts object will use the visNetwork method by default.

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

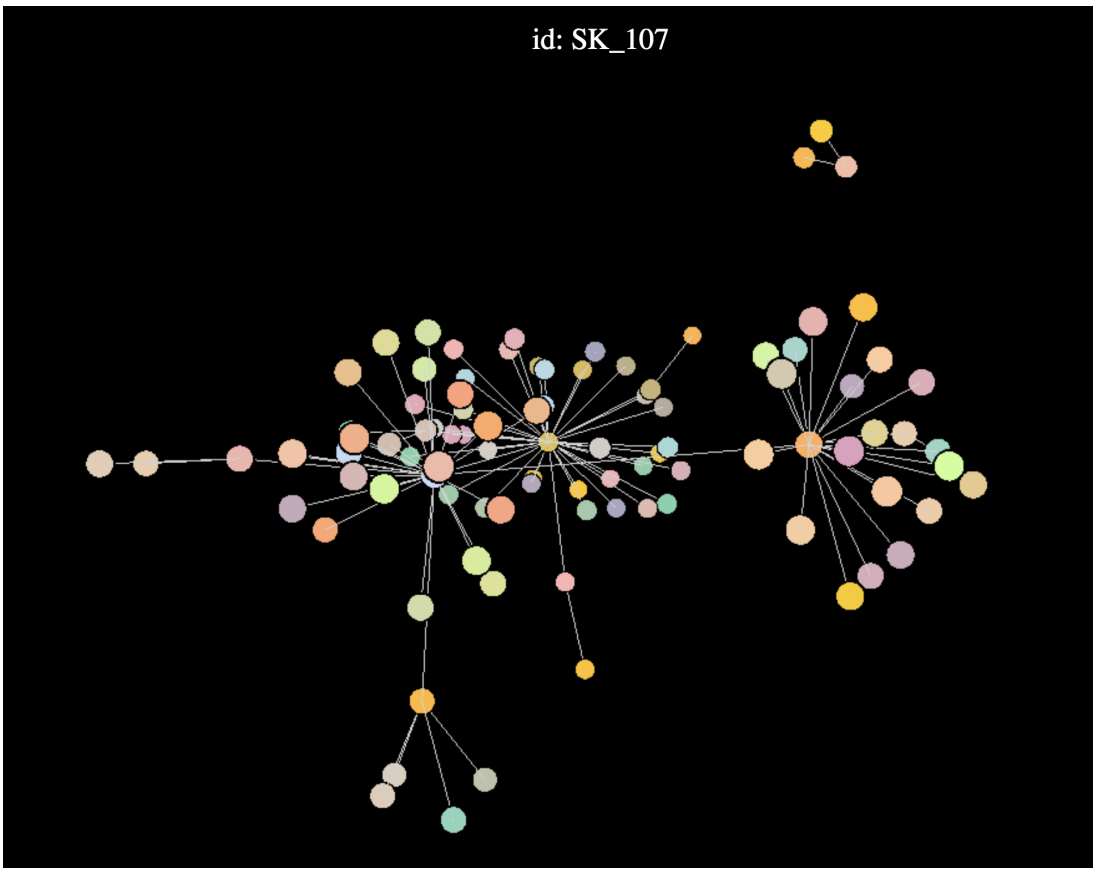

Figure 3. The graph3D() function generates a three-dimensional network plot.

```
# 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")
```

The `get_pairwise()` feature allows for specific analyses based on pairwise contact between individuals. With the contacts established, the function searches the linelist based on the supplied attribute. If the given column is a numeric or date object, `get_pairwise()` 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 (from and to) 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 linelist 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 string attribute, this can be helpful for tabulating information about different contact pairings by using the `table()` function.

```
# 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)
```

In terms of analysis, the package also leverages network techniques, including calculation of node degrees. This is implemented in `get_degree()`, which takes an `epicontacts` object and considers unique individuals across line lists and contacts as nodes. However, the function can be parameterized to only include cases from the line list if necessary. For directed networks the degree is available for "in", "out" or "both" directions per individual. The vector of "out" degrees estimates the reproductive number distribution[20].

```
# get degree for both
get_degree(x, "both")

# get degree out for only the contacts among individuals that appear in line list
get_degree(x, "out", only_linelist = TRUE)
```

Discussion

Benefits

While there are software packages available for epidemiological contact visualization and analysis, none aim to accommodate line list and contact data as purposively as `epicontacts`[21,22,23]. Furthermore, the package strives to solve a problem of plotting dense graphs by implementing interactive network visualization tools. A static plot of a network with many nodes and edges may be difficult to interpret. However, by dragging, hovering, zooming or rotating an `epicontacts` visualization, a user may be able to better understand the data. To the authors' knowledge there are no comparable visualization tools for disease outbreak networks. With that in mind, the fact that the functionality is freely packaged, with source code available, is beneficial to public health officials responding to outbreaks, epidemiological researchers as well as developers looking to incorporate or extend the package's functions in novel ways.

Implications for new tools

As with other open-source tools, the authors expect not only that the package will be used for data analysis, but also that it may be further developed and incorporated into new tools. The source code is available and contributions from other developers are welcome via Github pull requests. There is at least one R package that currently extends the original `epicontacts` functionality. The package `dibbler` for analyzing foodborne illness outbreak data builds upon the `epicontacts` structure to make its own similar data type[24].

Future considerations

`epicontacts` is a dynamic resource. Its maintainers have an eye towards new features and extended functionality. One area of future development could involve performance enhancement for visualizing extremely dense networks. Generating the interactive plots of large graphs is resource intensive. Further optimization of these visualization functions would be ideal. Additionally, future attention may be directed towards inclusion of alternative visualization methods, like adjacency heatmaps for example.

Conclusions

`epicontacts` provides a unified interface in the R statistical computing language for researchers and public health professionals to process, visualize and analyze disease outbreak data. The package and its source are freely available in a stable release on CRAN, as well as a development release on Github. By providing functionality designed around line list and contact list data, the authors aim to enable more efficient epidemiological analyses.

Software availability

1. URL link to where the software can be downloaded from or used by a non-coder: `https://CRAN.R-project.org/package=epicontacts`
2. URL link to the author's version control system repository containing the source code: `https://github.com/reconhub/epicontacts`
3. Link to source code as at time of publication (*F1000Research* TO GENERATE)

4. Link to archived source code as at time of publication (*F1000Research* TO GENERATE)
5. 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.

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] K. M. Wu and S. Riley. Estimation of the Basic Reproductive Number and Mean Serial Interval of a Novel Pathogen in a Small, Well-Observed Discrete Population. *PLoS ONE*, 11(2):e0148061, 2016.

[21] 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.

[22] 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.

[23] 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.

[24] Thibaut Jombart. *dibbler: Investigation of food-borne disease outbreaks*, 2017. URL `https://CRAN.R-project.org/package=dibbler`. R package version 0.0-2.