# Review of R Packages for Networks Visualization

Juan J. Cuadrado-Gallego
Miguel A. Losada
Cristopher Calva
*Computer Science Department*
*Universidad de Alcalá*
Alcalá de Henares, Madrid, Spain
jjcg@uah.es

Thomas Leek
*College of Letters and Science*
*University of Wisconsin-Stevens Point*
Stevens Point, Wisconsin, USA
tleek@uwsp.edu

José Manuel Molina
Miguel Ángel Patricio
*Applied Artificial Intelligence Group*
*Universidad Carlos III de Madrid*
Colmenarejo, Madrid, Spain
molina@ia.uc3m.es

***Abstract*—In the contemporary research environment, networks and the related techniques of analysis are involved in multiple subjects and are the object of broad and deep research. Visualization has become a prominent element of data science techniques for the analysis of networks. The selection and use of adequate techniques and tools is a key factor in the analysis of network analysis. The environments that can be used to develop visualizations, as for the rest of data science analytics, can be divided into open source and proprietary, and the main environments for the first are R and Python. The advantages of those two environments are that the community is continuously developing new packages for R and libraries for Python, that make possible the execution of more specific and sophisticated tasks. Network visualizations are also in continuous development and new techniques or types of graphics are being published. This paper presents results of a research performed with the objective of realizing a review and analysis of existing R packages for networks visualization. The main objective is to facilitate the selection of appropriate technique and the best tool to carry it out.**

***Index Terms*—Data Science, Data Analytics, Data Visualization, Networks, Networks Visualization, R, R packages**

## I. Introduction

Data visualization display measured quantities by means of the combined use of points, lines, a coordinate system, numbers, symbols, words, shading, and color [1], and its main objective is give the observer an easier understanding of as much information as possible, on observed data or on relationships in observed or discovered data, in the shortest possible time. Following [2] the Data Science framework has five knowledge areas and Data visualization is obtaining an increasing role in all five, but for some, and especially for Data Analytics, the importance of data visualization is undoubted and becoming more significant.

Networks analysis deals principally with the identification of connections between the participants in the net. Different types of graphs for the representation of networks can be found in the literature [3]. Network analysis is of primary importance to Data Science Analytics and the application of visualization techniques is relevant to network analysis. Specifically, this analysis is widely used in social networks that allows describing, understanding and explaining social structures [4]. Some of those purposes within social networks are commonly analyzing the intention to vote for an upcoming election [5], detecting terrorist groups [6] or identifying groups of people that may affect public health from an epidemiological point of view [7].

There are different environments to perform Data Analytics, and specifically Data Visualization, one of the most used by the community is R. Specifically, R is ”GNU S”, a freely accessible language and environment for statistical computing and graphing that provides a wide variety of statistical and graphical techniques: linear and nonlinear modeling, statistical tests, time series analysis, classification, grouping , etc. It also has CRAN, which is a network of ftp and web servers around the world that stores identical and updated versions of code and documentation for R [8]. This paper presents a review of R packages that allow for the visualization of networks.

This review consists in two joint tasks: the identification of packages that give to users the ability to perform network visualizations for each type of graph defined to visualize networks and a comparison of the those packages taking into account the following five attributes:

- Installation. Measure the difficulty of the package installation.
- Learning. Measure the difficulty of learning how to use the package.
- Complexity. Measure the complexity of package functions.
- Personalization. Measure the level of customization of the graph that will be obtained.
- Documentation. Measure the amount of documentation available about the paper.

## II. Network Visualization Techniques

This section presents the features of the different defined network visualization techniques. The graphs presented are all that are included in the classification [3] plus other set of graphs identified by the authors.

### A. Arc Diagram

An Arc Diagram represents relations between data with an axis, horizontal or vertical, in which each different value of the data is represented by a point (a node). The relation between them is represented by an arc that starts in one of the two data nodes and ends in the other. The thickness of the arc represents the frequency of the relation's appearance. Different colors can be used for different arcs in order to improve the clarity and readability of the graph. The main problem of this kind of graph arise when too many nodes or arcs must be represented,

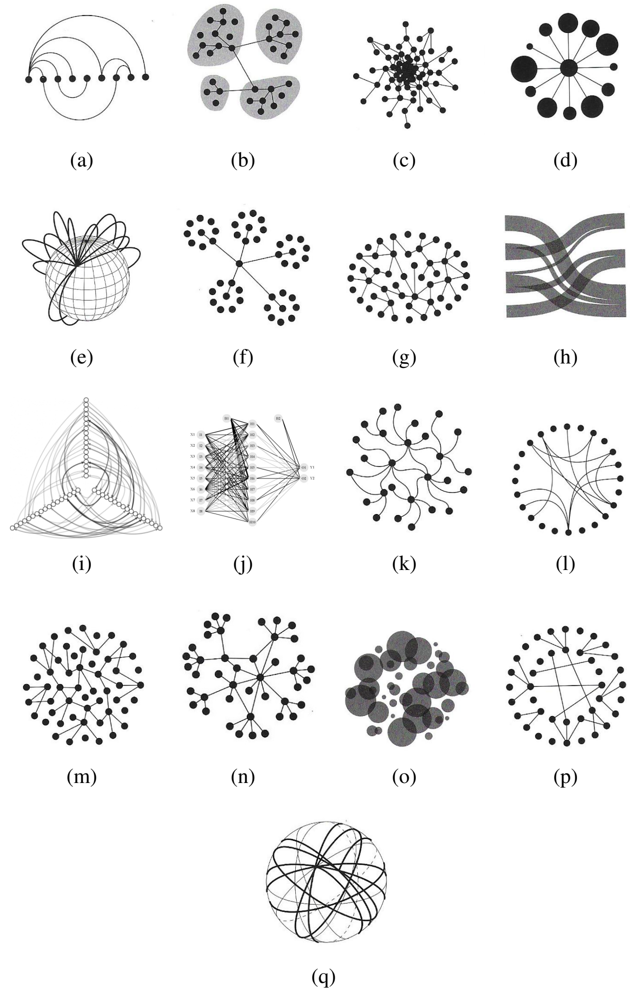


Fig. 1. (a) Arc Diagram (b) Area Grouping (c) Centralized Burst (d) Centralized Ring (e) Circled Globe (f) Circular Ties (g) Elliptical Implosion (h) Flow Chart (i) Hive (j) Neural Networks (k) Organic Rhizome (l) Radial Convergence (m) Radial Implosion (n) Ramification (o) Scaling Circles (p) Segmented Radial Convergence (q) Sphere. Partially taken from [3]

and, in consequence the readability of the the graph is seriously diminished.

### B. *Area Grouping*

Area Grouping is defined as the representation of sets or clusters of data, nodes, and how those clusters are related between them. The connections between nodes and cluster are done by lines. The different clusters are shadowed. This type of graphics allows the identification of communities or sub-trees densely connected inside a net diagram.

### C. *Centralized Burst*

Centralized Burst presents the data as nodes and the relation between them by lines. The position of each node in the graph is chaotic but the nodes with more connections are in the center, this is the reason of the name. This graphic allow the easy identification of the data with more connections as these are in the centre of the burst, that is, the center of the graphic.

### D. *Centralized Ring*

Centralized Ring represents the net starting from a central data node, from which lines come to the other nodes, related to the central one, located over a ring around the central node, with a concentric shape. The frequency of each data is represented by the width of the circle that represents the node. The nodes in the ring can also have relations between them and not only with the central node. This is a type of graphic used when the analysis of the network is focused over the relations that a specific datum, the central node, has with the rest of the data.

### E. *Circled Globe*

The Circled Globe graph has its name because the main element of the graph is a globe, that in many case represents the earth. The data nodes are often geographical points (cities, countries, etc.) and the lines represent the relations between them. The data nodes are on the surface of the sphere and the lines connecting them are also drawn out over the surface of the sphere.

### F. *Circular Ties*

The Circular Ties diagram is quite similar to a mix of the Area Grouping and the Centralised Ring. The shape is similar to the latter. But instead of having a single datum in each node, it displays clusters of data as the area grouping, but each cluster is presented by a centralized ring.

### G. *Elliptical Implosion*

The name implosion signifies that this graph imitates an inward explosion or collapse. The name elliptical means that the shape from which the implosion happen is an ellipse. Taking this into account this graph presents the nodes from the outside elliptical limit of the shape to the center, with a chaotic distribution of the internal nodes.

### H. *Flow Chart*

Flow Chart is a diagram in which the data, nodes, are located in vertical in the right and the left limits of the graph, and the nodes in the right are connected with the nodes in the left using curve lines, called lines of flux. The frequency of relations is represented by the width of the lines. Colors may also be used for improved clarity.

### I. *Hive*

This graph take its name from a Hive. The elements of the graph are three axis disposed each 120 grades in a circle, with one of them in the vertical. The nodes are disposed in those axis and the relations between them are solved with arcs.

### J. *Neural Network*

This graph is specific for drawing neural networks technique of data analysis. The nodes of the net in the different layers are disposed in vertical columns, one for each layer, and straight lines connect the different neurons or nodes in each layer.

### K. *Organic Rhizome*

An Organic Rhizome, that gives the name to this graph, is an underground stem with several buds growing horizontally and emitting roots and buds herbs from the nodes. Starting from this model, data nodes are the herb. Circular lines connect related nodes with the same shape.

### L. Radial Convergence

In a Radial Convergence graph, data nodes, are distributed over a circumference and the relations are drawn using curved lines connecting related data nodes across the circle. The frequency of relations is represented by the width of the lines. Colors may also be used for improved clarity.

### M. Radial Implosion

The Radial Implosion graph is exactly the same graph as an Elliptical implosion in reference to the means of developing the graph and the elements of the graphic. The only difference is that the external limit of the graph is circular, not elliptical.

### N. Ramification

As its name indicates, this network depiction is based on distributing its nodes through branches, in a manner similar to a *decision tree*, the nodes have child nodes and these in turn have more children, until one arrives at a node with no further relationships. So it becomes a leaf node because it only has a relationship with its single, father node.

### O. Scaling Circles

The main characteristic of the Scaling Circles diagram is that it emphasizes the importance of each data node through the use of smaller or larger circles depending on the factor, i.e. frequency. Colors may be used for improved clarity.

### P. Segmented Radial Convergence

In segmented radial convergence graphs, there are three main elements that makes these diagrams complex and with more semantic significance. The elements are nodes, represented by circles, relations, represented by straight lines, and segments that are represented by different concentric circles. Connections can be established between data in the same or different circles.

### Q. Sphere

This graph is similar to the circled globe, but with one main difference. The data nodes are on the surface of the sphere but the lines connecting them are also on the surface of the sphere, not over it.

## III. R Networks Visualization Tools

This section presents the identified packages of R that allow the user to deal with each one of the network visualization graphs identified in the previous section; the section also presents a comparison between the packages identified for each technique, taking into account the defined package attributes from the visualization point of view as defined in the introduction. Of course, in the case of the techniques for which only one package has been identified, this comparison does not appear. Also there are some networks graphs (Circular Ties, Radial Implosion, Segmented Radial Convergence and Sphere), for which no package of R has been identified that allows for this representation.

At the end of the section a table summarizes all the identification and comparative analysis of tools performed for all the network graphs.

### A. Arc Diagram

The identified R packages that allow for the production of arc diagrams are three:

- *arcDiagram* [9]. Developed only to make this kind of diagram. It can be used alone or as a plugin for the *igraph* package.
- *network* [10]. Developed with a general purpose to deal with network representations. Allows the realization of complex and dispersed representation, but for arc diagrams it must be used in conjunction with arcDiagram, since by itself it is unable to draw them.
- *ggraph* [11]. Developed as an extension of the *ggplot2* package, because its graphics grammar was not suitable for networks due to its reliance on tabular data entry.

The comparison between those packages taking the five variables of analysis is as follows:

- Installation. It is easier for ggraph and network because both of them are in CRAN and can be easily installed simply using the funtion *install packages*. However arcDiagram must be download and installed from *GitHub* from the author's webpage.
- Learning. arcDiagram is a fairly simple package to learn, since with just one function we can trace our network. Network and ggraph are more difficult to learn because more than one function must be used to draw the diagram, and for network, another package must be learned.
- Complexity. arcDiagram is simple, since its arcplot function is very easy to use and in few time we can have the layout of our network ready. From the network package we do not use functions and from ggraph, the complexity of its functions is a bit greater since it depends on the use of various functions to draw a relatively good and complex graph.
- Personalization. arcdiagram is quite good since it is sufficient to use the parameters of the arcplot function to trace the network as desired. It allows users to choose the color of the arcs, nodes, their sizes, ordering, etc. ggraph also offers a number of ways to trace the network.
- Documentation. ggraph and network is good and detailed being from CRAN packages, whereas with arcDiagram we have less detail.

### B. Area Grouping

The identified R packages that allow to perform Area Grouping are two:

- *igraph* [12]. Its main objective is to provide a set of data types and functions to implement graphic algorithms, quickly handle large graphics with millions of vertices and edges, and to allow for rapid prototyping using high-level language.
- *Linkcomm* [13]. Based on link communities, it provides a set of tools to generate, view, and analyze link communities on networks of arbitrary size and type. It also includes tools to generate, visualize, and analyze Overlapping Cluster Generator (OCG) communities.

The comparison between those packages taking the five variables of analysis is as follows:

- Installation. Very simple in both, since they belong to the CRAN repository.
- Learning. Igraph is easier to use, since it is not necessary to over-process the data, while Linkcomm incorporates more complex concepts such as OCG.
- Complexity. Higher in Lincomm, since it requires pre-treatment of the data, while in igraph the tracing functions are easier and more direct to use
- Personalization. Linkcomm is better in part because of its greater complexity in using it, giving you more depth. Still, igraph also has some tools that allow us to customize the visualizations.
- Documentation. As both packages belong to the CRAN repository, their documentation is very good.

### C. Centralized Burst

The identified R packages that allow for the development of Centralized Burst are two:

- *ggraph* [11]. Previously defined (see section 3 subsection A).
- *networkDynamic* [14]. It provides simple interface routines to facilitate the handling of network objects with complex timeless data. It is part of the set of statnet packages for network analysis.

The comparison between those packages taking the five variables of analysis is as follows:

- Installation. Very simple in both, since they belong to the CRAN repository.
- Learning. A little more complicated in ggraph when it comes to understanding how to draw networks, whereas networkDynamic uses only one function.
- Complexity. Although ggraph uses more functions to map the network, whereas networkDynamic uses a single function, the latter is more difficult to understand.
- Personalization. Greater in ggraph, since it has more algorithms and layouts for this purpose.
- Documentation. As both packages belong to the CRAN repository, their documentation is very good.

### D. Centralized Ring

The identified R package that allows for the production of a Centralized Ring graph is:

- *igraph* [12]. Previously defined (see section 3 subsection B).

### E. Circled Globe

The identified R package that allow to perform Circled Globe is:

- *Threejs* [15]. Provides real-time interactive 3D globe and scatterplots using three.js and the R htmlwidgets package.

### F. Elliptical Implosion

The identified R package that allows for the production of an Elliptical Implosion graph is:

- *qgraph* [16]. It provides tools for the visualization and analysis of weighted networks, as well as the calculation of the Gaussian graphic model. It also provides an interface to visualize data through network modeling techniques.

### G. Flow Chart

The identified R packages that allow for the production of a Flow Chart are two:

- *networkD3* [17]. Provides tools to create D3 networks, Javascript, as well as trees, dendrograms and Sankey Diagram from R.
- *plotly* [18]. Create interactive web graphics from ggplot graphics and/or a custom plotly.js JavaScript library interface inspired by graphing graphics.

The comparison between those packages taking the five variables of analysis is as follows:

- Installation. Very simple in both, since they belong to the CRAN repository.
- Learning. It is easier in networkD3 where we only use a function to trace the network and the parameters are easy to understand. On the other hand, plotly although it also needs a single function to trace the network, the parameters and the pre-treatment of the data is more complex.
- Complexity. It is simpler in networkD3, since, plotly requires a user to specify more exhaustively the data and the way in which the user wants to graph the network.
- Personalization. Very similar in the two packages. Despite the fact that the plotly parameters and the way to use it is more complex, in this case it does not imply that it has a higher degree of customization.
- Documentation. As both packages belong to the CRAN repository, their documentation is very good.

### H. Hive

The identified R package that allow to perform Hive is:

- *HiveR* [19]. Creates and plots 2D and 3D hive plots. Hive plots are a unique method of displaying networks of many types in which node properties are mapped to axes using meaningful properties rather than being arbitrarily positioned.

### I. Neural Networks

The identified R package that allow for the production of an Neural Network is:

- *nnet* [20]. It allows to implement advance neural networks with a single hidden layer, and for linear logarithmic multinomial models.

### J. Organic Rhizome

The identified R package that allow for the production of an Organic Rhizome graphic is:

- *igraph* [12]. Previously defined (see section 3 subsection B).

### K. Radial Convergence

The identified R packages that allow for the production of a Radial Convergence graph are three:

- *ggraph* [11]. (see section 3 subsection A).
- *igraph* [12]. (see section 3 subsection B).
- *circlize* [21]. It provides an implementation for the generation of circular layouts e, as well as an improvement of the

available software. The flexibility of the package is based on the use of low-level graphics functions, so that users can easily implement self-defined high-level graphics for specific purposes.

The comparison between those packages taking the five variables of analysis is as follows:

- Installation. Very simple in all three, since they belong to the CRAN repository.
- Learning. With the previous analysis, it is concluded that igraph is easier to use than ggraph, but the latter in turn is less complex than circlize. Therefore, the order from least difficulty to greatest is: igraph, ggraph and circlize.
- Complexity. As for learning difficulty, the order from least complex to most complex is: igraph, ggraph, and circlize. The reasons are the same as those seen for the previous characteristic.
- Personalization. In this case, customization is directly related to the complexity of the functions. So the package with the highest degree of customization is circlize, while igraph is the least and ggraph is somewhere in between.
- Documentation. Since all three packages belong to the CRAN repository, their documentation is very good.

### *L. Ramification*

The identified R packages that allow for the production of a Ramification graphic are two:

- *igraph* [12]. Previously defined (see section 3 subsection B).
- *networkD3* [17]. Previously defined (see section 3 subsection G).

The comparison between those packages taking the five variables of analysis is as follows:

- Installation. Very simple in both, since they belong to the CRAN repository.
- Learning. igrpah is quite easy to understand, since it has simple functions and implements others that allow using random network generation algorithms. On the other hand, networkD3 allows to trace networks in a very simple way.
- Complexity. It is very low in both, although igraph incorporates a series of algorithms that are necessary to know and may slightly increase their complexity.
- Personalization. Very similar for dost pauqetes. With igraph, you can combine certain functions to map more complex networks, but the fact that networkD3 networks are interactive, places both packages in a very similar position.
- Documentation. As both packages belong to the CRAN repository, their documentation is very good.

### *M. Scaling Circles*

The identified R packages that allow for the production of Scaling Circles are two:

- *ggraph* [11]. (see section 3 subsection A).
- *igraph* [12]. (see section 3 subsection B).

The comparison between those packages taking the five variables of analysis is as follows:

- Installation. Very simple in both, since they belong to the CRAN repository.
- Learning. Easier in igraph where only one function is needed to map and customize the network, while ggraph needs several.
- Complexity. ggraph requires further processing of the data and uses various functions to plot the networks. In contrast, igraph's functions are simpler for the same purpose.
- Personalization. Good for both packages, only ggraph implements networks in a different way, especially graphically.
- Documentation. As both packages belong to the CRAN repository, their documentation is very good.

TABLE I
R PACKAGES FOR NETWORK VISUALIZATION COMPARISON

| ***Package*** | ***I*** | ***L*** | ***C*** | ***P*** | ***D*** |
|---|---|---|---|---|---|
| **Arc Diagram** | | | | | |
| ArcDiagram | L[a] | VL | L | VH | M |
| network | VL | M | M | M | VH |
| ggraph | VL | M | M | VH | VH |
| **Area Grouping** | | | | | |
| igraph | VL | L | L | M | VH |
| Linkcomm | VL | M | M | H | VH |
| **Centralized Burst** | | | | | |
| ggraph | VL | M | L | H | VH |
| networkDynamic | VL | L | M | M | VH |
| **Centralized Ring** | | | | | |
| igraph | VL | VL | L | VH | VH |
| **Circled Globe** | | | | | |
| Threejs | VL | L | M | M | VH |
| **Elliptical Implosion** | | | | | |
| qgraph | VL | L | M | M | VH |
| **Flow Chart** | | | | | |
| networkD3 | VL | VL | L | M | VH |
| plotly | VL | M | H | M | VH |
| **Organic Rhizome** | | | | | |
| igraph | VL | VL | L | VH | VH |
| **Radial Convergence** | | | | | |
| ggraph | VL | L | L | M | VH |
| igraph | VL | VL | VL | L | VH |
| circlize | VL | M | M | H | VH |
| **Ramification** | | | | | |
| igraph | VL | VL | L | VH | VH |
| networkD3 | VL | VL | L | H | VH |
| **Scaling Circles** | | | | | |
| igraph | VL | VL | L | H | VH |
| ggraph | VL | L | M | H | VH |

a. VH: Very High; H: High; M: Medium; L: Low; VL: Very Low.

Table [I] summarizes the information presented on the analysis of the visualization tools. It is important to note that for the Installation, Learning and Complexity (I, L and P) fields the value Very High (VH) has a negative connotation meaning that the package in question is difficult to install, learn or understand its functions, while for the Personalization and Documentation fields, that same value takes a positive connotation, meaning that it has a high degree of personalization or documentation.

## IV. Conclusions

This paper has presented the results of research carried out with the main double objective to identify and analyze the available R packages to deal with the tasks to develop visualizations for networks. This have been done taking as an starting point a classification of networks done under the feature of their shape, this classification has taken one published that have been improved with the inclusion of new types.

Twenty different packages, some of them repeated, has been identified, for preforming eleven different types of networks, and all of them has been analyzed using five attributes that allow to know which of them could be more interesting to use for each problem to solve.

This work have been the first step for a more deep research that pretends to study the tools for data visualization in a more broad sense, not only for networks graphs but for all kind of data visualization diagrams, with all possible shapes and both, static and dynamic graphs, being these last ones, those graphs that only have sense when are visualized in movement; and that more broad sense means also the analysis of all the possible tools from all the main environments used in Data Science, that means the study of Python visualization tools, from the open source point of view, and also then main proprietary tools, and to identify those last ones the Gartner Magic Quadrant will be used.

## V. Acknowledgments

This research work have been carried out under the support of the Art. 83 contract 2019/00224/001 signed between the University of Alcalá and Ultra Tendency for research in Data Science and Big Data.

## References

[1] Edward R. Tufte. *The Visual Display of Quantitative Information*. Chesire, Connecticut: Graphics Press USA, 2001. ISBN: 9780961392147.

[2] Cuadrado-Gallego and Y. J. Demchenko, eds. *The Data Science Framework: a view form the Edison project*. Springer, 2020.

[3] Manuel Lima. *Visual Complexity: Mapping Patterns of Information*. Princeton Architectural Press, 2013. ISBN: 1616892196.

[4] Ulrik Brandes and Dorothea Wagner. "Analysis and visualization of social networks". In: *Graph drawing software*. Springer, 2004, pp. 321–340.

[5] Collins Udanor, Stephen Aneke, and Blessing Ogechi Ogbuokiri. "Determining social media impact on the politics of developing countries using social network analytics". In: *Program* (2016).

[6] C. C. Yang and T. D. Ng. "Terrorism and Crime Related Weblog Social Network: Link, Content Analysis and Information Visualization". In: *2007 IEEE Intelligence and Security Informatics*. 2007, pp. 55–58.

[7] Nicholas A Christakis and James H Fowler. "Social network visualization in epidemiology". In: *Norsk epidemiologi= Norwegian journal of epidemiology* 19.1 (2009), p. 5.

[8] *The Comprehensive R Archive Network*. Accesed on 2020-07-20. URL: https://cran.r-project.org/.

[9] Gaston Sanchez. *arcdiagram*. 2014. URL: https://github.com/gastonstat/arcdiagram#readme.

[10] Carter T. Butts. "network: a Package for Managing Relational Data in R." In: *Journal of Statistical Software* 24.2 (2008). URL: http://www.jstatsoft.org/v24/i02/paper.

[11] Thomas Lin Pedersen. *ggraph: An Implementation of Grammar of Graphics for Graphs and Networks*. R package version 2.0.3. 2020. URL: https://CRAN.R-project.org/package=ggraph.

[12] Gabor Csardi and Tamas Nepusz. "The igraph software package for complex network research". In: *InterJournal* Complex Systems (2006), p. 1695. URL: http://igraph.org.

[13] Alex T Kalinka and Pavel Tomancak. "linkcomm: an R package for the generation, visualization, and analysis of link communities in networks of arbitrary size and type". In: *Bioinformatics* 27 (2011), pp. 2011–2012. DOI: 10.1093/bioinformatics/btr311.

[14] Carter T. Butts, Ayn Leslie-Cook, Pavel N. Krivitsky, and Skye Bender-deMoll. *networkDynamic: Dynamic Extensions for Network Objects*. R package version 0.10.1. 2020. URL: https://CRAN.R-project.org/package=networkDynamic.

[15] B. W. Lewis. *threejs: Interactive 3D Scatter Plots, Networks and Globes*. R package version 0.3.3. 2020. URL: https://CRAN.R-project.org/package=threejs.

[16] Sacha Epskamp, Angélique O. J. Cramer, Lourens J. Waldorp, Verena D. Schmittmann, and Denny Borsboom. "qgraph: Network Visualizations of Relationships in Psychometric Data". In: *Journal of Statistical Software* 48.4 (2012), pp. 1–18. URL: http://www.jstatsoft.org/v48/i04/.

[17] J.J. Allaire, Christopher Gandrud, Kenton Russell, and CJ Yetman. *networkD3: D3 JavaScript Network Graphs from R*. R package version 0.4. 2017. URL: https://CRAN.R-project.org/package=networkD3.

[18] Carson Sievert. *plotly for R*. 2018. URL: https://plotly-r.com.

[19] Bryan A. Hanson. *HiveR: 2D and 3D Hive Plots for R*. R package version 0.3.63. 2020.

[20] W. N. Venables and B. D. Ripley. *Modern Applied Statistics with S*. Fourth. ISBN 0-387-95457-0. New York: Springer, 2002. URL: http://www.stats.ox.ac.uk/pub/MASS4.

[21] Zuguang Gu, Lei Gu, Roland Eils, Matthias Schlesner, and Benedikt Brors. "circlize implements and enhances circular visualization in R". In: *Bioinformatics* 30 (19 2014), pp. 2811–2812.