Journal of Statistical Software

May 2020, Volume 93, Issue 11. doi: 10.18637/jss.v093.i11

Network Coincidence Analysis: The netCoin R Package

Modesto Escobar
Universidad de Salamanca

Luis Martinez-Uribe
Universidad de Salamanca

Abstract

The aim of the R package **netCoin** is to explore data structures using a number of statistical techniques that share the handling of interdependent variables. The main objective of this analysis is to detect events, characters, objects, attributes or characteristics that tend to appear together within a given set of scenarios. Its most notable feature is the combination of traditional multivariate statistical analysis and network analysis supported by topological graph theory. In addition, **netCoin** produces HTML graphs using the **D3.js** visualization library to support the interactive exploration of networked data. Among its many applications, **netCoin** can be used for the analysis of multiple responses in questionnaires to explore relevant associations, for the development of textual networks, for the study of ecological communities, for audience analysis, for mining large databases or for basket market analysis.

Keywords: co-occurrence analysis, social network analysis, multivariate analysis, interactive graphs.

1. Introduction

netCoin (Escobar, Barrios, Prieto, and Martinez-Uribe 2020) is an R package which performs network coincidence analysis, whose aim is to find out the structure and the degree to which a series of events (subjects, objects or characteristics) tends to occur together within certain limits called scenarios. To discover these patterns, this package generates visualizations of the coincidences through interactive network graphs via a web browser.

Graphs represent elements (nodes) that may or may not be connected (edges). Coincidence graphs consist of two types of information: a set of nodes or vertices (events), $N = (n_1, n_2, \ldots, n_J)$, and a set of lines, links or edges (coincidences), $L = (l_1, l_2, , \ldots, l_L)$ (Wasserman and Faust 1994).

The interactive web graphs produced by **netCoin** allow modification of their elements and their features (such as size, color or position). In addition, data about nodes and edges is displayed below the graph, and both the data and the graphs can be downloaded onto every computer connected to the graph via an Internet browser.

Among the several interactive elements available, the following are key:

a. The label, size, color and shape of the events or nodes based on their properties. It is also possible to represent groups of nodes with similar characteristics as well as using images to depict them.

b. The width, weight, color and any text of the edges that represent the coincidences between the events based on the edges' properties, such as their frequency, degree of coincidence or statistical significance.

c. Nodes can be filtered manually or dynamically by either the value of their attributes or their connections.

d. Edges can also be filtered dynamically by the value of their attributes.

The starting point for these graphs is the incidence matrix, made up of two dimensions: The rows that contain the scenarios where the coincidences are to be detected and the columns which contain the elements whose coincidences are of interest for this particular study.

An application for this analysis arises from the complexity of working with multiple-choice questionnaires. To illustrate this use, we may consider a simple question about job hunting strategies that unemployed people might use to find a job. For this question, a survey could include responses such as family, friends, sending résumés to companies, job ads in newspapers or job centers. Therefore, what is the best way to code and save this information? One column is insufficient, as we might be dealing with multiple alternative answers.

However, two solutions might be applied: firstly, using as many columns as there are possible responses. The question may ask respondents to provide their top three job hunting strategies. In this case, three columns could be enough, providing a different strategy in each one. Nonetheless, if the number of responses is open, the number of columns needed could be codified using the multiple mode (from 1 to 5 in the case of 5 possible responses) or in a dichotomous fashion using one column for each response and codifying them as one for those selected and as zero for those not selected.

Another use for the network coincidence analysis is content analysis. A survey, apart from multiple-choice questions, may also include open-ended questions. The text from the responses to those open-ended questions may be divided into words or phrases whose coincidences may also be the subject of analysis. Plenty of specialized software could be used (N-Vivo, Atlas-Ti, QDA Miner, MaxQDa) whose main objective is to enable the classification of large text corpora such as transcripts of focus groups and interviews. In addition to this, algorithms are emerging to perform thematization (Corman, Kuhn, McPhee, and Dooley 2002; Blei, Ng, and Jordan 2003; Feinerer, Hornik, and Meyer 2008; Van Attenveld 2008; Grimmer and Stewart 2013; Roberts *et al.* 2014; Lucas, Nielsen, Roberts, Stewart, Storer, and Tingley 2015) and sentiment analysis (Young and Soroka 2012), which could use graph representation.

Due to **netCoin**'s core objective to produce graphs, it turns this into a fit-for-purpose methodology for the study of bimodal networks, which present two sets not internally connected but

interconnected between them. One of the typical applications of these structures are affiliation networks, which represent the connections between an actor with a set of social situations (Wasserman and Faust 1994). For example, bimodal networks could help study the membership of an executive group in a company or the events that the inhabitants of a certain village attend.

Those relationships can be studied using bipartite graphs or hypergraphs as well as dual hypergraphs, although in most cases the representation of only one of the sets is of interest (such as the actors or the inhabitants in the previous examples), and any bimodal network can be transformed into a unimodal one, which leads to the preference for the co-participation matrix. Precisely, the main operation behind **netCoin** is generating a coincidence matrix (unimodal) from an incidence matrix (bimodal) to convert the former into a graph.

Another application of network coincidence analysis is the study of species within different ecosystems. The coexistence of bird species in the Galápagos Islands is one highly popular case among biologists (Sanderson 2000). For this study, a probabilistic co-occurrence method based on the hypergeometric distribution, which is also included in **netCoin**, has been developed (Veech 2013).

2. Similar software

There are a variety of tools within the statistics and data analysis domain that perform similar operations to **netCoin** to visually analyze the structure of binary data.

It is also possible to find common ground with machine learning techniques, especially the association rules (Borgelt 2012) that have binary matrices as their starting point. In contrast, **netCoin** focuses on the associations between pairs of events while `apriori()` and `eclat()` procedures seek for higher order connections available through the package **arules** (Hahsler, Grün, and Hornik 2005).

The comparative qualitative analysis (QCA; Ragin 1987, 2000) has a similar input, i.e., a matrix made up of zeros and ones, although it is based on different algorithms using Boolean logic. Coincidence analysis (Baumgartner 2009) is derived from the QCA and they both have R packages: **QCA** (Wickham and Miller 2019) and **cna** (Baumgartner and Thiem 2015).

It is also worth mentioning some packages associated with text analysis, like **tm** (Feinerer 2019; Feinerer and Hornik 2019), **RTextTools** (Jurka, Collingwood, Boydstun, Grossman, and van Atteveldt 2014), **textometry** (Loiseau, Vaudor, Decorde, and Heiden 2015), **lda** (Chang 2015), **stm** (Roberts, Stewart, and Tingley 2019a,b) and **tidytext** (Robinson and Silge 2020).

Tools with a specific focus on network analysis and visualization include four major packages: **igraph** (Csardi and Nepusz 2006; Csardi 2020), **network** by Butts (2008, 2019), the graphical complement **networkD3** (Grandrud, Allaire, and Rusell 2016; Allaire, Grandrud, Rusell, and Yetman 2015) and **visNetwork** by Almende, Benoit, and Titouan (2019). The first two are powerful tools for analyzing networks and can represent them in a non-interactive way, unless they are used in conjunction with **tcltk2** (Grosjean 2014), but they lack the analytic instruments to study coincidences and the ability to create HTML graphs. Another similar package is **RJSplot** (Prieto and Barrios 2017), which produces interactive and dynamic graphics widely used in DNA structure data analysis. The last three are more similar to **netCoin**. However, they lack statistical tools to produce the coincidence graphs. Outside of the R environment, a variety of social network analysis tools exist such as **Gephi** (Bastian,

Heymann, and Jacomy 2009), **Pajek** (Batagelj and Mrvar 1998) or the Phyton (Van Rossum *et al.* 2011) package **NetworkX** (Schult and Swart 2008).

It is important to mention those packages which specialize in co-occurrence in community structures. Griffith, Veech, and Marsh (2016a) created the **cooccur** package (Griffith, Veech, and Marsh 2016b), with incidence matrices similar to those of **netCoin**, but only using the hypergeometric distribution. They refer to other packages to detect pairs of species that share some space with one another such as **picante** (Kembel *et al.* 2010), **spaa** (Zhang 2016) and **vegan** (Oksanen *et al.* 2019).

In terms of similarity and distance calculations, packages like **stats** (R Core Team 2020), **proxy** (Meyer and Buchta 2019) and even **parallelDist** (Eckert 2018) cover most of the metrics that **netCoin** calculates. However, they do not include some of the coincidence analysis metrics needed, such as frequency, conditional frequency or statistical significance. In addition to this, **netCoin** allows the calculation of more than one metric at the same time just by calling one function. This reduces calculation time and thus improves performance.

A similar package to **netCoin** is **qgraph** (Epskamp, Cramer, Waldorp, Schmittmann, and Borsboom 2012; Epskamp, Costantini, Haslbeck, and Isvoranu 2020), which provides an interface to visualize data through network modeling techniques. However, **qgraph** is intended to represent a correlation matrix or a factor analysis statically, while **netCoin** is specialized in the representation of qualitative variables transformed into dichotomies and its parameters can be interactively changed through a web page.

In sum, despite the fact that there are many packages and software tools to analyze binary metrics and represent networks, **netCoin** adds value by providing the possibility to efficiently calculate a series of distance and similarity measures, including their statistical significance, and allowing the generation of interactive graphic output in HTML.

3. Coincidence analysis

Co-occurrences have been widely studied in many fields, especially in the content analysis of texts (Carley 1993; Lund and Burgess 1996; Popping 2000, 2003; Matsuo and Ishizuka 2004) and in the study of ecological communities (Diamond and Gilpin 1982; Connor and Simberloff 1983; Veech 2013). In addition to this, there is extensive literature that focuses on applications and many R packages that facilitate their analysis as seen in the previous section.

netCoin focuses on a particular form of dealing with co-occurrences, which is called coincidence analysis, and whose aim is to detect which people, subjects, objects, attributes or events tend to appear at the same time in different limited spaces (Diaconis and Mosteller 1989; Baumgartner 2009; Escobar 2015).

An event (j) is a potential outcome of a random experiment. The set of possible outcomes is called a sample space and is composed of a series of elementary mutually exclusive events.

A scenario (i) is each one of the results of a complex experiment made up of a set of events (X_j) with varying degrees of dependence between each other. A scenario can also be defined as a spatial and temporal set in which the researcher collects information on the events that take place.

Since the events of the scenarios are not mutually exclusive, they can be represented using

Scenarios	Head	Tail
I	1	0
II	1	1
III	1	1
IV	0	1

Table 1: Incidence matrix with 4 scenarios after tossing 2 coins.

Scenarios	Head	Tail
I	2	0
II	1	1
III	1	1
IV	0	2

Table 2: Occurrence matrix with 4 scenarios after tossing 2 coins.

	Head	Tail
Head	3	
Tail	2	3

Table 3: Coincidence matrix of the 4 scenarios of 2 coins.

dichotomous vectors (they can either occur or not) or vectors containing natural numbers (number of times each event occurs in a given scenario).

Therefore, the set of observed n scenarios can be represented as an *incidence matrix* ($\mathbf{I} = (x_{ij})$). In one dimension (generally the rows) the matrix contains the scenarios (i) and in the other dimension (commonly the columns) it contains the events (j). This matrix consists of 1s and 0s indicating if the events occurred or not, respectively, within the scenario. Alternatively, the occurrence matrix, which records the number of appearances of the event in every scenario, can be employed.

This distinction will be better understood with this simple example: If two coins are tossed four times, each toss represents a scenario where the events heads and tails are of interest. The three possible results for each toss of the two coins are: a) two heads, no tails; b) a head and a tail, and c) two tails and no head. The incidence matrix can be presented as shown in Table 1. On the other hand, the occurrence matrix must reflect the two heads or two tails obtained when the result is not head and tail (see Table 2).

Coincidence and co-occurrence matrices can be calculated from the incidence and occurrence matrices.

Definition. *Two coincident events (j and k) are those which occur together in the same scenario i.*

$$(x_{ij} > 0 \wedge x_{ik} > 0) \Rightarrow f_{ijk} = 1$$

Along with the basic coincidence in a given scenario i, when considering whether two events coincide in a multiple set of scenarios, the total number of coincidences of the events j and k can be obtained.

$$f_{jk} = \sum_{i=1}^{I} f_{ijk}$$

In addition, we can distinguish different degrees of coincidences. Thus, the most basic coincidence classification would distinguish between:

a. **No coincidence**: Two events that never occur in the same scenario, i.e., they are mutually exclusive ($f_{jk} = 0$).

b. **Simple coincidence**: Two events are merely coincident if they occur together in at least one scenario ($f_{jk} > 0$).

c. **Total coincidence**: Two events that always occur together in the same scenarios. If one of them occurs, then the other does too ($f_{jk} = f_{jj} = f_{kk}$). A special case is the subtotal coincidence in which the other event occurs only if the first occurs and not vice versa ($f_{jk} = f_{jj} > f_{kk}$), i.e., the occurrence of the more frequent event (k) does not necessarily imply the occurrence of the less frequent event (j).

From the incidence matrix, the *coincidence matrix* $\mathbf{F} = (f_{ij})$ can be calculated using this expression: $\mathbf{F} = \mathbf{I}^\top\mathbf{I}$. This is an example of how to project a bimodal network to a unimodal one. The elements of this matrix are either univariate (f_{jj}) or bivariate (f_{jk}) frequencies of the different events in the set of scenarios (i) contained in the rows of $\mathbf{I}$.

From the coincidence matrix ($\mathbf{F}$) three probabilistic measures can be derived:

a. The ***marginal probability*** of X_j, denoted as $\mathsf{P}(X_j)$, can be obtained by dividing the frequencies of each event (f_{jj}) by the total number of scenarios (n) in which it could have occurred:

$$\mathsf{P}(X_j) = \frac{f_{jj}}{n}.$$

b. The ***joint probability*** of two events X_j and X_k, expressed as $\mathsf{P}(X_{jk})$ is given by the frequency of occurrence in the same scenario divided by the set of scenarios considered in a given set:

$$\mathsf{P}(X_{jk}) = \frac{f_{jk}}{n}.$$

c. The ***conditional probability***, denoted as $\mathsf{P}(X_j|X_k)$, expresses the possibility that a certain event occurs when the second event has already occurred. It is obtained by dividing the joint probability by the marginal probability of the conditional event:

$$\mathsf{P}(X_j|X_k) = \frac{\mathsf{P}(X_{jk})}{\mathsf{P}(X_k)} = \frac{f_{jk}}{f_{kk}}.$$

With the conditional probability, we can create a coincidence gradient, the ***probable coincidence***, between two events when their conditional probability is greater than 50%:

$$\mathsf{P}(X_j|X_k) > 0.5.$$

When working with samples of scenarios instead of the whole universe, the upper limit of the confidence interval can be estimated under the alternative hypothesis of $\mathsf{P}(X_j|X_k) < 0.5$ using the formula

$$L_{\text{sup}} = \frac{f_{jk}}{f_{kk}} + \frac{t_{\alpha, f_{kk}-1}}{2\sqrt{f_{kk}}},$$

Type of coincidence	Definition	Asymmetric	Statistical test
Null	$f_{jk} = 0$	No	No
Simple	$f_{jk} > 0$	No	No
Probable	$f_{jk}/f_{kk} > 0.5$	Yes	Yes
Conditional	$f_{jk} > f^*_{jk}$	No	Yes
Subtotal	$f_{jk} = f_{jj} < f_{kk}$	Yes	No
Total	$f_{jk} = f_{jj} = f_{kk}$	No	No

Table 4: Types of coincidences.

where $t_{\alpha,f_{kk}-1}$ is the value of the Student distribution for $f_{kk} - 1$ degrees of freedom with a significance level of α.

The ***conditional coincidence*** is another coincidence gradient. It is derived from the concept of independence of events. Two events are independent if Equation 1 is true:

$$\mathsf{P}(X_j) = \mathsf{P}(X_j|X_k) \iff \frac{f_{jj}}{n} = \frac{f_{jk}}{f_{kk}}. \tag{1}$$

Therefore, for that condition to be met, the following condition needs to be verified:

$$f^*_{jk} = \frac{f_{jj} f_{kk}}{n}.$$

From this equation, two events have a conditional coincidence when their frequency is greater than the expected (f^*_{jk}) under the assumption of independence:

$$f_{jk} > \frac{f_{jj} f_{kk}}{n} = f^*_{jk}.$$

It is also known (Haberman 1973) that the difference between f_{jk} and f^*_{jk} assumes asymptotically a normal distribution with the following standard error:

$$\sqrt{f^*_{jk}(1 - f_{jj}/n)(1 - f_{kk}/n)},$$

which can be used to standardize (r_{jk}) the difference between the empirical frequency of coincident events (f_{jk}) and the expected frequency (f^*_{jk}) under the assumption of mutual independence:

$$r_{jk} = \frac{f_{jk} - f^*_{jk}}{\sqrt{f^*_{jk}(1 - f_{jj}/n)(1 - f_{kk}/n)}}.$$

For small samples, the one-sided Fisher exact test, which employs the hypergeometric distribution should be used instead (Fisher 1935; Finney 1948).

The degrees of coincidence that can be detected between each pair of events is summarized in Table 4.

3.1. Coincidence metrics

In addition to classifying coincidences into different types, they can be measured using binary proximity metrics (Hubálek 1982; Gower 1985). These measures have a maximum value of

	Event X_k	
Event X_j	Present	Absent
Present	a	b
Absent	c	d

Table 5: Contingency table.

one when there is total coincidence between two dichotomous events and a value of 0 when there is total independence between them. Some of them can take negative values, in which case the minimum value could be -1 when two incompatible events are implied.

For the calculation of these metrics each element (f_{jk}) of the coincidence matrix can be split into the following system equivalences:

$$
\begin{aligned}
a &= f_{jk} \\
b &= f_{jj} - f_{jk} \\
c &= f_{kk} - f_{jk} \\
d &= n - f_{jj} - f_{kk} + f_{jk}
\end{aligned}
$$

Therefore, for each pair of events, Table 5 can be elaborated. With these four figures (a, b, c, d), representing the frequencies of the four states of presence/absence of two events in the set of scenarios studied, binary proximity measures are obtained.

These coefficients or binary proximity metrics can be classified into four types: The *first* one includes metrics that are similar to that of *matching* (Rogers and Tanimoto 1960; also known as Rogers and Tanimoto). They are the result of divisions with a numerator with both positive coincidences (the two events occur in the same scenario) and negative coincidences (the two events are absent in the same scenario), and a denominator where all scenarios are considered with different weights. The metrics belonging to this category are *Rogers* (Rogers and Tanimoto 1960), *Sneath* (Sneath and Sokal 1962), *Anderberg* (1973) and *Gower* (1985). These measurements should be used when considering coincidence both when two events are present in the same scenario, as well as when both are not present.

$$
\begin{aligned}
\mathit{Matching} &= \frac{a+d}{a+b+c+d} \\
\mathit{Rogers} &= \frac{a+d}{(a+d)+2(b+c)} \\
\mathit{Sneath} &= \frac{2(a+d)}{2(a+d)+(b+c)} \\
\mathit{Anderberg} &= \left(\frac{a}{a+b}+\frac{a}{a+c}+\frac{d}{c+d}+\frac{d}{b+d}\right)/4 \\
\mathit{Gower} &= \frac{ad}{\sqrt{(a+b)(a+c)(d+b)(d+c)}}
\end{aligned}
$$

In the *second* type of metrics there is *Jaccard* (1901). Here, scenarios where neither of the two events whose coincidence degree we intend to measure (d) are excluded. Therefore, neither the numerator nor the denominator include those scenarios without any of the two events.

Metrics of this type also include *Dice* (Jaccard 1901), *Antidice* (Anderberg 1973), *Ochiai* (1957) and *Kulczynski* (1927). In this case, events that are not present in the same scenario are not considered to be coincident, and only those scenarios where at least one event has occurred are coincident.

$$
\begin{aligned}
Jaccard &= \frac{a}{a+b+c} \\
Dice &= \frac{2a}{2a+b+c} \\
Antidice &= \frac{a}{2+2(b+c)} \\
Ochiai &= \frac{a}{\sqrt{(a+b)(a+c)}} \\
Kulczynski &= (\frac{a}{a+b} + \frac{a}{a+c})/2
\end{aligned}
$$

The *third* type of similarity metrics for binary data only includes *Russell and Rao* (1940). It only considers those scenarios to be similar in which both events occur. It excludes from the numerator those in which none of the events occurs, considering that this does not indicate that the scenarios are similar. However, unlike the similarity metrics such as *Jaccard*'s, all the possible scenarios are present in the denominator of the equation. This coincidence measure only takes into account coincident events and contemplates all scenarios, including those in which both events are not present. Logically, if there are no scenarios where neither of the two is present, then both are equal. However, if within an infinite number of scenarios neither of the two events existed, the value of Russell and Rao would be zero, while Jaccard would be 1 by convention.

$$
Russell\ and\ Rao = \frac{a}{a+b+c+d}
$$

Finally, in the *fourth* type we may include all metrics in which frequencies of coincidences (whether the events occur or not) are compared (subtracted) with frequencies of no coincidences (scenarios where an event occurs but the other one does not). Thus, these measurements can be positive if coincident events predominate, or negative otherwise, i.e., when the scenarios in which the events do not coincide predominate. Metrics of this type include *Pearson* (1900), *Yule* (1900) and *Hamann* (1961). This modality is similar to the correlation coefficients and has the advantage of presenting both positive and negative values. Positive values imply that whenever an event is present, the other is as well; while negative ones evidence that in most cases, the presence of an event implies the absence of the other.

$$
\begin{aligned}
Pearson &= \frac{ad-bc}{\sqrt{(a+b)(a+c)(b+d)(c+d)}} \\
Yule &= \frac{ad-bc}{ad+bc} \\
Hamann &= \frac{(a+d)-(b+c)}{a+b+c+d}
\end{aligned}
$$

All the previous expressions are called similarity metrics. To turn them into distance measurements, the following expression can be used $distance = 1 - similarity$. If the metric has a range between 0 and 1, then these limits are preserved, although with a different meaning, as

Type	Measures (abbreviation for procedures)
Frequencies	Frequencies (f), Relative frequencies (x), Conditional frequencies (i, ii)*
Degrees	Coincidence degree (cc), Probable degree (cp)
Expected values	Expected (e), Confidence interval (con)
Matching	*Matching* (m), *Rogers* (t), *Gower* (g), *Sneath* (s), *Anderberg* (and)
Jaccard	*Jaccard* (j), *Dice* (d), *antiDice* (a), *Ochiai* (o), *Kulczynski* (k)
Russell	*Russell* (r)
Pearson	*Pearson* (p), *Haberman* (h), *Yule* (y), *Hamann* (ham), *odds ratio* (od)
Probabilistic	p value of Haberman (z), hypergeometric p greater value (hyp)

Table 6: Similarity measures (* i: conditioned by the source frequency; ii: conditioned by the target frequency).

the 0 indicates complete coincidence. Nevertheless, if the metric range is between -1 and $+1$, the new similarity metric will be between 0 and 2, with 1 indicating complete independence and higher values meaning that two events coincide less often than by mere chance.

An outline of these measures and the abbreviations to obtain them with **netCoin** can be found in Table 6.

3.2. Adjacency matrix

Coincidence and distance matrices have been covered. Both types can be transformed into adjacency matrices. An adjacency matrix connects each pair of events depending on whether their coincidence metric is above a certain value. Thus, it is a square matrix with as many rows and columns as the number of events being studied, and formed by elements representing the number of coincidences between every pair of events. Using all the previous metrics, adjacency matrices can be formed in the following ways:

a. With the simple coincidences so that there will be a connection between two events provided that they have coincided in a single scenario.

Frequency matrix

	odd	even	small	large
odd	54			
even	0	46		
small	41	13	54	
large	13	33	0	46

Adjacency matrix

	odd	even	small	large
odd	–			
even	0	–		
small	1	1	–	
large	1	1	0	–

b. With total or subtotal coincidences so that two completely overlapping events will be connected. In the first category, it will be a symmetrical connection, and in the case of subtotal coincidences, it will only connect the less frequent category and the most frequent ones.

Conditional frequencies

	odd	even	small	large
odd	100.0			
even	0.0	100.0		
small	75.9	28.3	100.0	
large	24.1	71.7	0.0	100.0

Adjacency matrix

	odd	even	small	large
odd	–			
even	0	–		
small	0	0	–	
large	0	0	0	–

c. With the probable or conditional coincidences, connecting events with more than 50% probability in the first case and a positive residual (r_{jk}).

Standardized residuals (r_{jk})

	odd	even	small	large
odd	100.0			
even	−10.0	100.0		
small	4.8	−4.8	100.0	
large	−4.8	4.8	0.0	100.0

Adjacency matrix

	odd	even	small	large
odd	–			
even	0	–		
small	1	0	–	
large	0	1	0	–

d. With the statistical tests applied to the probable or conditional coincidences, in which case we could have statistically significant coincidences with different degrees or levels of significance (0.05, 0.01, 0.001, 0.0001, ...).

Significance of r_{jk}

	odd	even	small	large
odd	–			
even	1.0e+00	–		
small	3.2e–06	1.0e+00	–	
large	1.0e+00	3.2e–06	1.0e+00	–

Adjacency matrix

	odd	even	small	large
odd	–			
even	0	–		
small	1	0	–	
large	0	1	0	–

e. With the coincidence metrics, in which case one of the 14 possible coincidences must be chosen, setting a threshold (0.50, for instance) from which it can be considered that two events are coincident.

Jaccard's similarity

	odd	even	small	large
odd	1.00			
even	0.00	1.00		
small	0.61	0.15	1.00	
large	0.15	0.56	0.00	1.00

Adjacency matrix

	odd	even	small	large
odd	–			
even	0	–		
small	1	0	–	
large	0	1	0	–

3.3. Layouts

The same way that a series of coincidences can become an adjacency matrix, the latter can be converted into a graph. As previously said, a graph $\mathcal{G}$ consists of "two sets of information: a set of nodes (events), $\mathcal{N} = \{n_1, n_2, \ldots, n_g\}$, and a set of lines (coincidences), $\mathcal{L} = \{l_1, l_2, \ldots, l_L\}$ between a pair of nodes"(Wasserman and Faust 1994).

Layout	Argument	Abbreviation
Random disposition of vertices	`"layout.random"`	`"ra"`
Rectangular grid disposition	`"layout.grid"`	`"gr"`
Circle distributed vertexes	`"layout.circle"`	`"ci"`
Star disposition of vertices	`"layout.star"`	`"st"`
Fruchterman and Reingold	`"layout.fruchterman.reingold"`	`"fr"`
Kamada and Kawai	`"layout.kamada.kawai"`	`"ka"`
Forced directed layout (GEM)	`"layout.gem"`	`"ge"`
Simulated annealing algorithm	`"layout.davidson.harel"`	`"da"`
Multidimensional scaling coordinates	`"layout.mds"`	`"md"`
Tidy arrangement of vertices	`"layout.reingold.tilford"`	`"re"`
Layered directed acyclic graphs	`"layout.sugiyama"`	`"su"`
Large scale graphs	`"layout.drl"`	`"dr"`
Large graph layout	`"layout.lgl"`	`"lg"`

Table 7: Layout algorithms. References: fr, Fruchterman and Reingold (1991); ka, Kamada and Kawai (1989); ge, Frick *et al.* (1995); da, Newman (2006); md, Cox and Cox (2001); re, Reingold and Tilford (1981); su, Sugiyama *et al.* (1981); dr, Martin *et al.* (2008); lg, Martin *et al.* (2008).

An additional problem is where to draw each node, i.e., the spatial distribution of the nodes. Thanks to **igraph**, **netCoin** can be laid out according to the criteria in Table 7.

If none of these layouts are indicated, **netCoin** uses a dynamic Fruchterman-Reingold algorithm by default. Alternatively, the user can provide a matrix with two columns indicating the coordinates of those nodes that are going to be fixed in the representation. Leftover nodes should be stated as `NA` and would be placed according to a forced directed mechanism.

3.4. Communities

Cluster analysis is "a set of methods for constructing a (hopefully) sensible and informative classification of an initially unclassified set of data, using the variable values observed on each individual" (Everitt 2003). In agglomerative hierarchical clustering methods, there are various procedures to join cases using dendrograms: single, complete, average, median, Ward, etc. In the coincidence analysis, clustering could be useful to classify events according to their concurrences, using the Haberman residuals (r_{jk}) or another distance matrix (geodesic, matching, Jaccard, ...) as inputs to the clustering method.

Events j and k are structurally equivalent if, for all events, $l = 1, 2, \ldots, g\,(l \neq j, k)$, and for all associations $r = 1, 2, \ldots, R$, event j has a relation to l if and only if k also has a relation to l. Consequently, structurally equivalent events are those that have identical edges with the rest of events. Structural equivalence can imply "community", but it does not have to (e.g., if each community consists of a standard set of hierarchical actors), and community does not have to imply structural equivalence. Events can be partitioned into subsets of structural equivalence using a *hierarchical clustering* or a similar algorithm of classification. **netCoin** allows us to obtain the **igraph** procedures listed in Table 8.

Community	Argument	Abbreviation
Edge-betweenness	`"cluster_edge-betweenness"`	`"ed"`
Fast-greedy	`"cluster_fast_greedy"`	`"fa"`
Label propagation	`"cluster_label_prop"`	`"la"`
Leading eigenvector	`"cluster_leading_eigen"`	`"le"`
Louvain	`"cluster_louvain"`	`"lo"`
Optimal modularity	`"cluster_optimal"`	`"op"`
Sping glass	`"cluster_spinglass"`	`"sp"`
Walktrap	`"cluster_walktrap"`	`"wa"`

Table 8: Communities algorithms. References: ed, Girvan and Newman (2002); fa, Wakita and Tsurumi (2007); la, Raghavan *et al.* (2007); le, Newman (2006); lo, Blondel *et al.* (2008); op, Good *et al.* (2009); sp, Reichardt and Bornholdt (2006); wa, Pons and Latapy (2006).

4. The R package netCoin

Some of **netCoin**'s statistical and graphical features were originally implemented in Stata (StataCorp 2019) as the **coin** ado program (Escobar 2015). This initial Stata program lacked the graphical interactivity which provides agile data exploratory capabilities. That is the main reason why R was chosen to generate an extended version of the original **coin** program.

Firstly, the **shiny** (Chang, Cheng, Allaire, Xie, and McPherson 2020) and **igraph** packages were used to achieve graph results, but what provided the solution to accomplish the desired interactivity was the integration with the **D3.js** data visualization library (Bostock, Ogievetsky, and Heer 2011). In addition to this, R code has been written to obtain the coincidence metrics and their significance.

4.1. Installation

The **netCoin** package is available from the Comprehensive R Archive Network (CRAN) at `https://CRAN.R-project.org/package=netCoin` and has dependencies on three other R packages **igraph** (Csardi and Nepusz 2006), **Matrix** (Bates and Mächler 2019) and **haven** (Dusa and Thiem 2020) which are loaded with **netCoin**.

```
R> install.packages("netCoin")
R> library("netCoin")
```

4.2. Overview with three simple examples

The **netCoin** package incorporates every coincidence analysis element detailed in Section 3. The functions included help the analyst convert the data into an incidence matrix that is suitable for the analysis, produce the coincidence matrix, calculate all the statistical indicators, generate the nodes and edges of the graph, produce interactive network visualizations and export those networks as '`igraph`' objects.

The basic input is an incidence binary matrix, which can be obtained with the function `dichotomize()` in case of absence. This function can be applied to both character variables and factor variables. In addition to this, among the former it is able to split fragments separated by a constant chain, whose default value is the null character (`""`).

Argument	Meaning
`sep = ""`	The separator in case that the variables are composed.
`min = 1`	Minimum frequency of the value of a variable to be considered as an event.
`length = Inf`	Maximum number of events to be considered.
`values = NULL`	Events to be converted into dichotomies (not for multiple composed variables).
`sparse = FALSE`	Produce a sparse matrix instead of a data frame.
`add = TRUE`	Add the new columns to the original data frame.
`sort = TRUE`	Order the new columns by their frequencies.

Table 9: Arguments of function `dichotomize`.

In addition to the data frame and the variable or variables to be dichotomized, the arguments of this function are given in Table 9.

The simplest example can be applied to the `dice` data frame included in the package:

```
R> data("dice", package = "netCoin")
R> events <- dichotomize(dice, "dice", add = FALSE, sort = FALSE)
R> head(events)

   1 2 3 4 5 6 dice:None
V1 1 0 0 0 0 0         0
V2 0 1 0 0 0 0         0
V3 0 0 0 0 1 0         0
V4 0 0 0 1 0 0         0
V5 0 1 0 0 0 0         0
V6 0 0 0 0 1 0         0
```

Thus, a new data frame with 6 columns corresponding to the six possible events of throwing a dice would be obtained.

We would have to add the argument `sep =` in case of factor variables composed of several events. As a second example, imagine that we tossed two coins in unison ten times into the air. The results could be `"H,H"`, `"T,H"`, `"H,T"`, `"T,T"`, each with the same probability. Therefore, to convert the events of each toss into elementary events, we use `dichotomize()` with the argument `sep = ","`.

```
R> set.seed(10)
R> coins <- data.frame(coin = cut(runif(10), c(0, 0.25, 0.50, 0.75, 1),
+    labels = c("H,H", "T,H", "H,T", "T,T")))
R> events <- dichotomize(coins, "coin", sep = ",")
R> events

    coin H T coin:None
V1   H,T 1 1         0
V2   T,H 1 1         0
```

```
V3   T,H 1 1         0
V4   H,T 1 1         0
V5   H,H 1 0         0
V6   H,H 1 0         0
V7   T,H 1 1         0
V8   T,H 1 1         0
V9   H,T 1 1         0
V10  T,H 1 1         0
```

Once we have an incidence matrix, we obtain a '`coin`' object, a list composed by the number of events and the coincidence matrix, with the function `coin()`. Then, the function `edgeList()` converts a '`coin`' object into a data frame containing an edge list with the similarity measures stated in the procedure argument. By default, `edgeList()` produces Haberman residuals with their p values. The third example considers the presence of three people (`"Man"`, `"Woman"` and `"Undet."`) in four different scenarios.

```
R> frame <- data.frame(A = c("Man; Woman", "Woman; Woman", "Man; Man",
+    "Undet.; Woman; Man"))
R> data <- dichotomize(frame, "A", sep = "; ")[2:4]
R> coin <- coin(data)
R> coin

n= 4
       Man Woman Undet.
Man      3
Woman    2     3
Undet.   1     1      1

R> edges <- edgeList(coin)
R> edges

  source target  Haberman         Z
3    Man Undet. 0.6666667 0.2707349
6  Woman Undet. 0.6666667 0.2707349
```

Finally, the function `netCoin()` can mix the nodes (extracted from the '`coin`' object) with the edge list data frames in order to produce a '`netCoin`' object, and if the argument `dir = "directory"` is used, a directory will be created with a graph within a web page whose main file is named `index.html`.

The '`netCoin`' object has three methods: `print()` shows a sample (until 6) of nodes and links with their attributes, `summary()` shows the basic statistics of the nodes, and `plot()` shows the corresponding graph in the computer's default browser.

```
R> nodes <- asNodes(coin)
R> netCoin(nodes, edges)
R> (net <- netCoin(nodes, edges))
R> print(net)
```

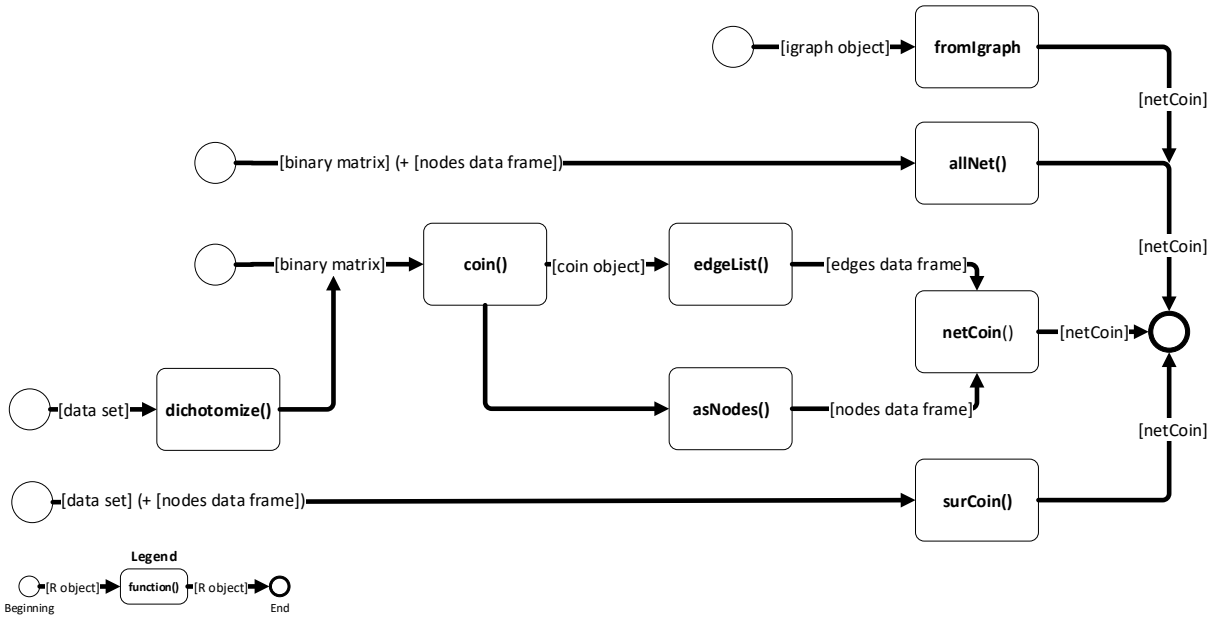

Figure 1: **netCoin** processes to create a graph.

```
Nodes(3):
         name frequency
Man       Man         3
Woman   Woman         3
Undet. Undet.         1

Links(2):
  source target  Haberman         Z
3    Man Undet. 0.6666667 0.2707349
6  Woman Undet. 0.6666667 0.2707349
```

```
R> summary(net)
```

```
3 nodes and 2 links.
frequency distribution of nodes:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  1.000   2.000   3.000   2.333   3.000   3.000
Haberman's distribution of links:
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
 0.6667  0.6667  0.6667  0.6667  0.6667  0.6667
```

```
R> plot(net)
```

Alternatively, the '`netCoin`' object could be obtained directly from a binary incidence matrix with the `allNet()` function, where more than 40 arguments can be controlled, although the only required argument is the incidence matrix. However, if we want to obtain the directory

with the graph, we must add the `dir = "directory"` argument. Other sources to obtain a '`netCoin`' object are an '`igraph`' object with the function `fromIgraph`, and a data set with factor variables with the function `surCoin()`. See the four functions that obtain a '`netCoin`' object in Figure 1.

```
R> frame <- data.frame(A = c("Man; Woman", "Woman; Woman", "Man; Man",
+    "Undet.; Woman; Man"))
R> data <- dichotomize(frame, "A", sep = "; ")[2:4]
R> allNet(data)
```

Using the previous `data` data frame, a set of coincidence measures and their significance can be printed with the `edgeList` function, whose input must be a '`netCoin`' object.

```
R> edgeList(coin(data),
+    proc = c("frequency", "Jaccard", "Pearson", "Haberman", "Z", "fisher"),
+    criteria = "fisher", max = 1)
```

```
  Source Target coincidences  Jaccard   Pearson  Haberman     p(Z) p(Fisher)
1    Man  Woman            2 0.500000 -0.333333 -0.666667 0.729265      1.00
2    Man Undet.            1 0.333333  0.333333  0.666667 0.270735      0.75
3  Woman Undet.            1 0.333333  0.333333  0.666667 0.270735      0.75
```

4.3. Other examples

Multigraph coincidence analysis with data of families from Renaissance Italy

The following example uses data about families from Renaissance Italy from Padgett and Ansell (1993). It consists of a data frame (families) with information about Italian families of the Renaissance, and another data frame (links) with the marriage and business bonds between families.

```
R> data("families", package = "netCoin")
R> data("links", package = "netCoin")
```

The previous `coin()`, `edgeList()`, `asNodes()` and `netCoin()` functions can be executed together with the `allNet()` function where several arguments can be specified (see Table 10).

Two networks are generated representing the business and marriage bonds between the two families with the following commands.

```
R> G <- allNet(incidence = links[links$link == "Marriage", -17],
+    nodes = families, layout = "md", criteria = "f", minL = 1,
+    size = "frequency", color = "seat",
+    main = "Marriage links between Italian families",
+    note = "Data source: Padgett & Ansell (1983)")
```

Argument	Meaning
`incidence`	A data frame that contains the incidence matrix.
`nodes`	A data frame with at least one vector of names.
`layout`	The algorithm selected for the network topology.
`criteria`	The statistical criteria to be used for the selection of the edges.
`minL`	Minimum value of the statistic to represent the edge in the graph.
`size`	Name of the vector with size in the nodes data frame.
`color`	Name of the vector with color variable in the nodes data frame.
`main`	Upper title of the graph.
`note`	Lower title of the graph.

Table 10: Arguments of function `allNet`.

Function	Description
`dichotomize`	Function to convert factor or character column(s) in a data frame into a set of dichotomous columns. Their names will correspond to the labels or text of every category.
`coin`	This function generates a '`coin`' object from an incidence matrix data frame. A '`coin`' object consists of a list with two elements: the number of scenarios, and a coincidence matrix of events, whose main diagonal figures are the frequency of events and outside the said diagonal there are conjoint frequencies of these events
`asNodes`	From a '`coin`' object, this function generates a data frame of nodes.
`edgeList` (`sim`)	Function to convert a coincidence matrix into an edge list calculating a variety of coincidence (proximity) metrics. The `sim` function produces the same information, but as a list of proximity matrices instead.
`netCoin`	The `netCoin` function produces an interactive '`netCoin`' object from two data frames: one including nodes with attributes, and another one containing edges also with its own attributes.
`multigraphCreate`	This function produces an interactive multinetwork with several '`netCoin`' objects.
`fromIgraph`	From an '`igraph`' object, this function generates a '`netCoin`' object.
`toIgraph`	With this function an '`igraph`' object is generated from a '`netCoin`' object.
`allNet`	Produces a '`netCoin`' object from a data frame or a matrix with dichotomous values.
`surCoin`	Produces a '`netCoin`' object from a data frame with factor variables accepting also a '`tbl_df`' class (see package **haven**).

Table 11: **netCoin** main functions.

```
R> H <- allNet(incidence = links[links$link == "Business", -17],
+    nodes = families, layout = "md", criteria = "f", minL = 1,
+    size = "frequencb", color = "seat",
+    main = "Marriage links between Italian families",
+    note = "Data source: Padgett & Ansell (1983)")
```

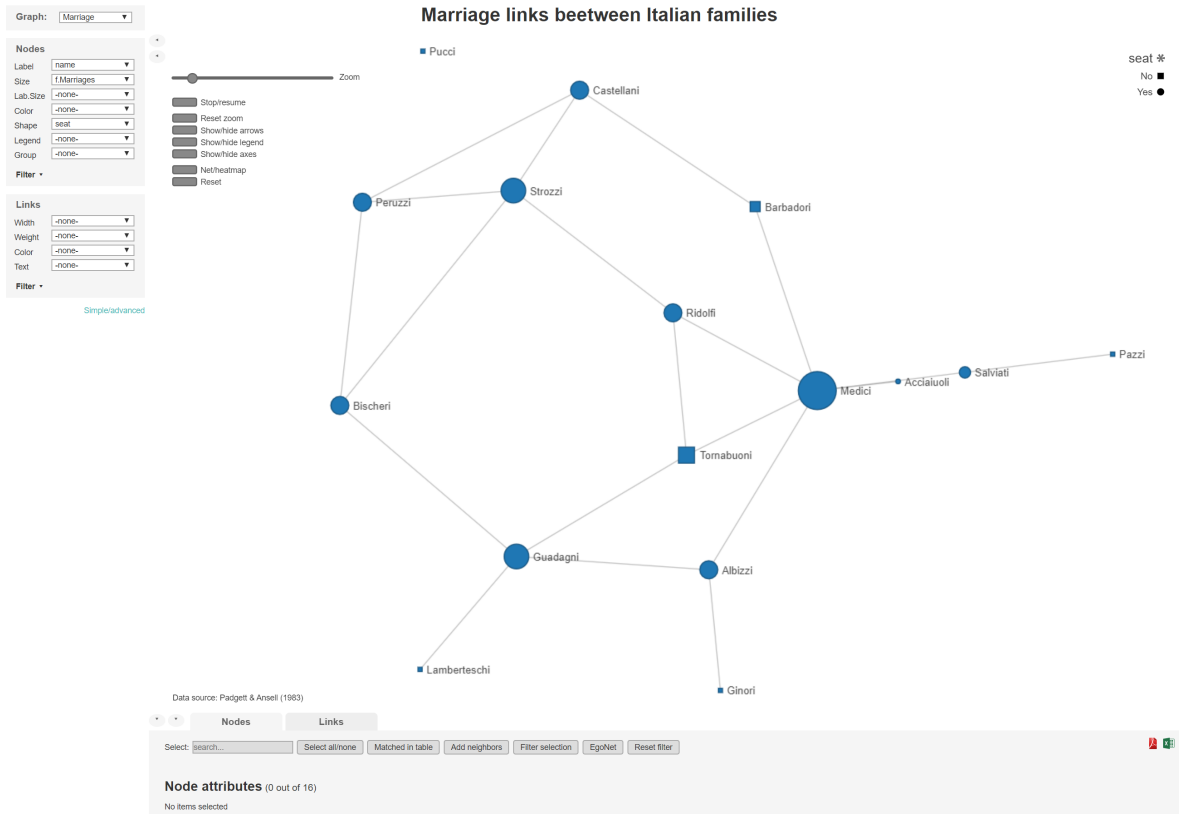

Figure 2: Graph of links between Italian families.

```
R> G

Title: Marriage links between Italian families

Nodes(16):
       name f.Marriages f.Business wealth priorates seat
 Acciaiuoli           1          0     10        53  Yes
    Albizzi           3          0     36        65  Yes
  Barbadori           2          4     55         0   No
   Bischeri           3          3     44        12  Yes
 Castellani           3          3     20        22  Yes
     Ginori           1          2     32         0   No
...

Links(20):
     source      target frequencies
    Albizzi   Guadagni           1
    Albizzi     Medici           1
    Albizzi     Ginori           1
 Acciaiuoli     Medici           1
  Barbadori Castellani           1
```

```
  Barbadori     Medici           1
...

Data source: Padgett & Ansell (1983)
```

The '`netCoin`' object `G` (as well as the non-shown `H`) is composed of two data frames. In the first (`nodes`) there are the families' attributes: frequency of marriage links (`f.Marriages`), frequency of business links (`f.Business`), a wealth index (`wealth`), number of priories held (`priorates`) and holding of at least one priorate (`seat`). In every row of the `links` data frame there are two families with a column indicating the existence of a link (coincidence) between them.

Once the two networks are ready, the function `multigraphCreate()` generates both graphs in the specified directory (see Figure 2).

```
R> multigraphCreate(Marriage = G, Business = H, dir = "italian")
```

Sanderson's analysis of species co-occurrences

This section uses one of the most renowned data examples in ecology. Charles Darwin compiled data about 13 species of finches and 17 of the Galápagos Islands (Sanderson 2000) on which they could be found.

We prepare the nodes' attributes (`finches`) and their incidences in the islands (`Galapagos`). Afterwards, we have to add the images in a specific directory in order to refer to them in the `allNet()` function.

```
R> data("Galapagos", package = "netCoin")
R> data("finches", package = "netCoin")
R> finches$species <- system.file("extdata", finches$species,
+    package = "netCoin")
```

Here, a few extra features are added to the graph shown in Figure 3:

- `criteria = "hyp"`: The statistical criteria to be used for the strength of the edges.
- `maxL = 0.05`: Maximum value of the statistic to include the edge in the list.
- `lwidth = "Haberman"`: Name of the vector with width variable in the links data frame.
- `lweight = "Haberman"`: Name of the vector with weight variable in the links data frame.
- `image = "file"`: Name of the vector with image files in the nodes data frame.
- `layout = "mds"`: The algorithm selected for the network topology.

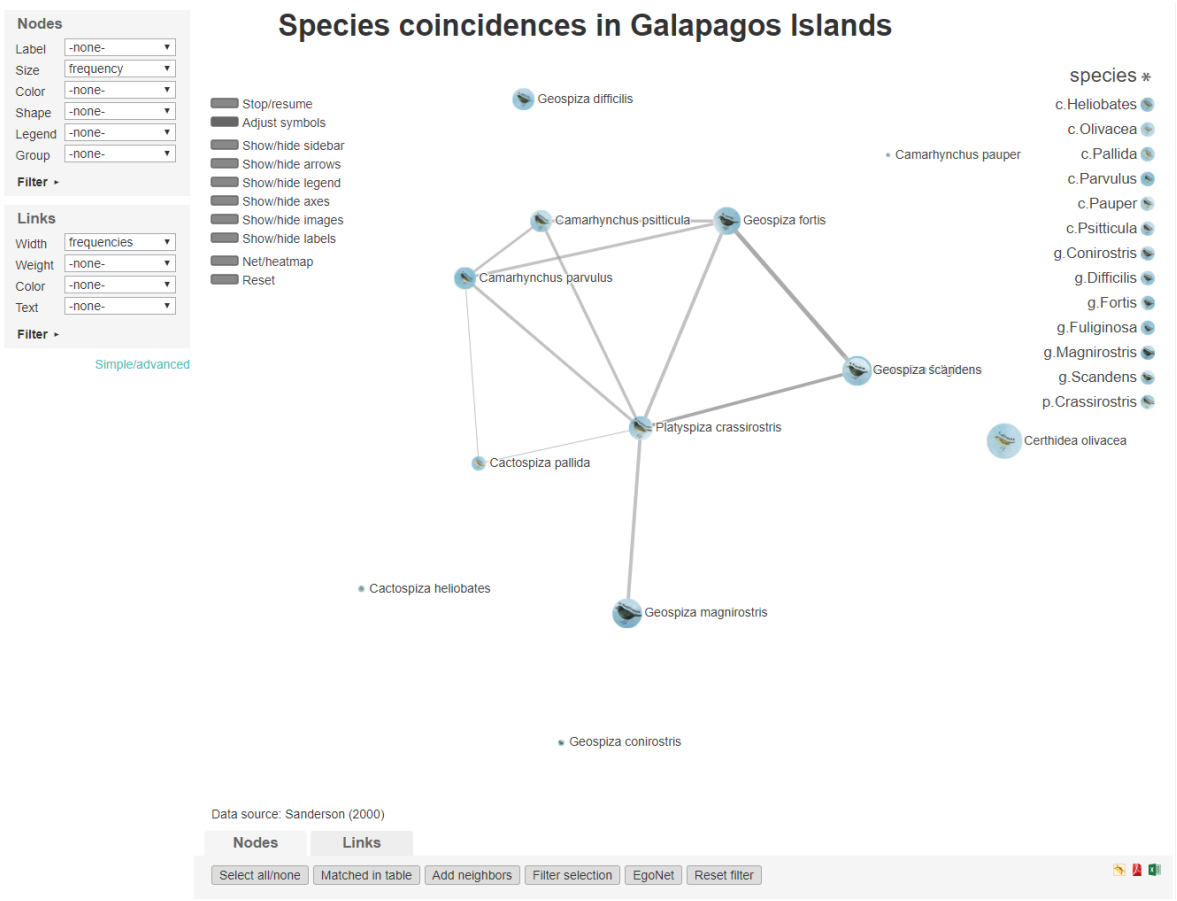

Figure 3: Graph of finches coincidences in Galápagos Islands.

```
R> Net <- allNet(Galapagos,
+    frequency = TRUE, procedures = "frequencies", criteria = "hyp",
+    layout = "mds", nodes = finches, maxL = 0.05, size = "frequency",
+    image = "species", lwidth = "frequencies", cex = 1.35, controls = 2:4,
+    main = "Species coincidences in Galapagos Islands",
+    note = "Data source: Sanderson (2000)")
R> Net

Title: Species coincidences in Galapagos Islands

Nodes(13):
                  name frequency        %     type
 Geospiza magnirostris        14 82.35294 Geospiza
       Geospiza fortis        13 76.47059 Geospiza
   Geospiza fuliginosa        14 82.35294 Geospiza
   Geospiza difficilis        10 58.82353 Geospiza
     Geospiza scandens        12 70.58824 Geospiza
  Geospiza conirostris         2 11.76471 Geospiza
...
```

```
Links(14):
                Source                    Target frequencies   p(Fisher)
 Geospiza magnirostris Platyspiza crassirostris          11 0.029411765
       Geospiza fortis      Geospiza fuliginosa          13 0.005882353
       Geospiza fortis        Geospiza scandens          12 0.002100840
       Geospiza fortis  Camarhynchus psitticula          10 0.014705882
       Geospiza fortis    Camarhynchus parvulus          10 0.014705882
       Geospiza fortis Platyspiza crassirostris          11 0.006302521
...

Data source: Sanderson (2000)
```

In this example, the only attributes of nodes are `frequency`, percentage (`%`) and `type`. The column `specs` has been suppressed because it is used to create the images from the images file names. More importantly, the links attributes are 1) `frequencies`, for example the number of coincidences of source and target finches, and 2) `p(Fisher)`, which is the error probability of rejecting the one-side alternative hypothesis, in case that it is true that two species are not coincident on each island (scenario).

Once the '`netCoin`' object is ready, the function `plot()` generates its graphical representation in a temporary directory (see Figure 3), or in the directory specified in the `dir` argument. In this way, all the necessary files to be deposited in a web server are saved so that anyone can view them and interact with them using a browser.

```
R> plot(Net)
```

Graphical comparison of two networks

netCoin can also be used to graphically compare networks of co-occurrences. For instance, the previous graph of the Galápagos Islands finches (`Net`) can be compared with a random null model obtained from the same data with the function `cooc_null_model()` of the **EcoSimR** package (Gotelli, Hart, and Ellison 2015). Among the possibilities offered by this program, we opted for the nullity of co-occurrences and the Sim9 algorithm, which is a sequential swap (Gotelli 2000; Strona, Nappo, Boccacci, Fattorini, and San-Miguel-Ayanz 2014).

Once the theoretical or null model is randomly obtained (`nullData`), it could be analyzed and represented with the command `allNet()` assessing the significance of its co-occurrence links. Previously, in order to better compare the empirical data obtained by Darwin with the random null model data, the positions of the nodes of the null model are set using those of the empirical model. After using the hypergeometric distribution (`criteria = "hyp"`) and a level of significance of 0.05 (`maxL = 0.05`), the new graph (`NullNet`) only has two co-occurrences out of the possible 78 (paired combinations of 13 fiches).

To represent these two or more networks at the same time, the function `multigraphCreate()` is used with the parallel argument assigned as true. It can be observed (Figure 4) that the species are located in the same place and have the same size, proportional to their presence in the islands, but the number of links is much smaller, because they have been randomized and a filter of significance in the argument of the `allNet()` function has been set.

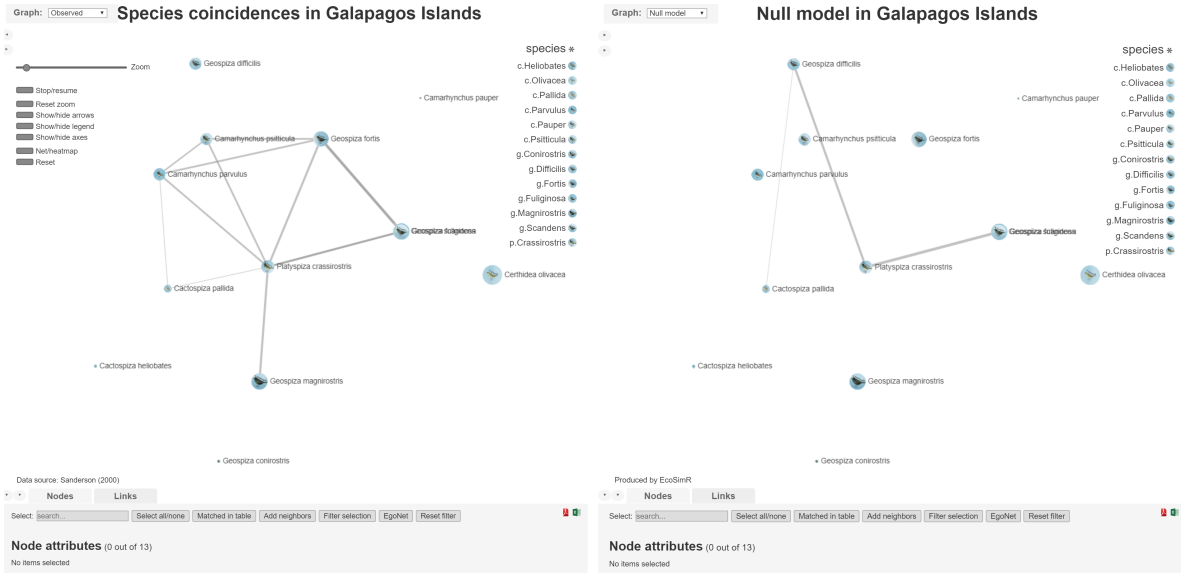

Figure 4: Graph of finches coincidences in Galápagos Islands.

```
R> library("EcoSimR")
R> layout.Net <- cbind(Net$nodes$fx, Net$nodes$fy)
R> set.seed(2016)
R> nullModel <- cooc_null_model(t(Galapagos), nReps = 1000, burn_in = 500,
+    algo = "sim9", metric = "checker")
R> nullData <- t(nullModel$Randomized.Data)
R> colnames(nullData) <- colnames(Galapagos)
R> NullNet <- allNet(nullData, frequency = TRUE, procedures = "frequencies",
+    criteria = "hyp", maxL = 0.05, layout = layout.Net, nodes = finches,
+    size = "frequency", image = "species", lwidth = "frequencies",
+    cex = 1.4, controls = 2:3, main = "Null model in Galapagos Islands",
+    note = "Produced by EcoSimR")
R> multigraphCreate("Observed" = Net, "Null model" = NullNet,
+    mode = "parallel")
```

Survey analysis

Another interesting use for **netCoin** is that of survey analysis applied to explore relationships between variables including those from multiple choice questions. The straightforward analysis shown below uses the package **haven** (Dusa and Thiem 2020) to read a SPSS (IBM Corporation 2017) survey demo file. Three variables are selected for the analysis: `gender`, `inccat` (income category in thousands) and `carcat` (primary vehicle price category).

The `plot()` function is applied to the result of the `surCoin()` function with those three variables as inputs. This produces the graph in Figure 5 where the male node is connected to the lowest and highest incomes as well as the economy and luxury vehicle categories. On the other hand, the female node is linked to income categories in the middle range and either the standard or the luxury vehicle price category.

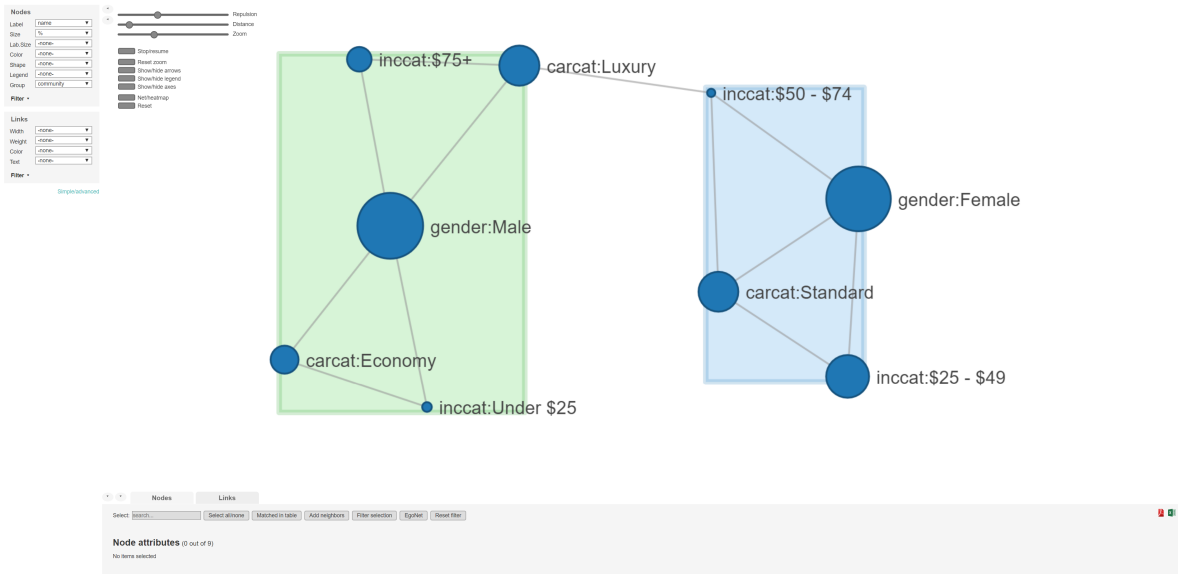

Figure 5: Graph of survey data multiresponse question.

```
R> library("haven")
R> survey <- read_spss(file = "demo.sav")
R> variables <- c("gender", "inccat", "carcat")
R> plot(surCoin(survey, variables, communities = "Louvain"))
```

4.4. Performance

To test the **netCoin** performance, several random datasets were generated with a different number of cases (1,000 and 50,000) and events (10, 50, 100). Tests were performed with six datasets: M(1,000× 10), M(1,000×50), M(1,000×100), M(50,000×10), M(50,000×50), M(50,000 ×100). Calculations for Jaccard were compared using **netCoin** and **parallelDist**. The results show faster times for **parallelDist** when the number of cases or events is smaller. But when the number of cells (cases times events) grows, then **netCoin** offers better results as shown by Figure 6. As time grows exponentially with the number of cells, time is represented by its logarithmic values in this figure.

The package produces interactive graphs that work well with up to 1500 edges. Using more than 1500 edges makes the interaction with the graph slow due to browser memory limitations.

5. Concluding comments

The **netCoin** package offers an opportunity for the interactive analysis and visualization of data sets composed of every kind of data insofar as variables are dichotomized. It contains a large variety of similarity measures to connect the events that co-occur in the same scenarios.

In order to select the relevant coincidences, **netCoin** incorporates two models of probability: the normal distribution through the Haberman residuals for a large number of scenarios, and the hypergeometric model for small data collections. Its main aim is to represent coincidences through a graph, which is particularly useful when many events are to be analyzed.

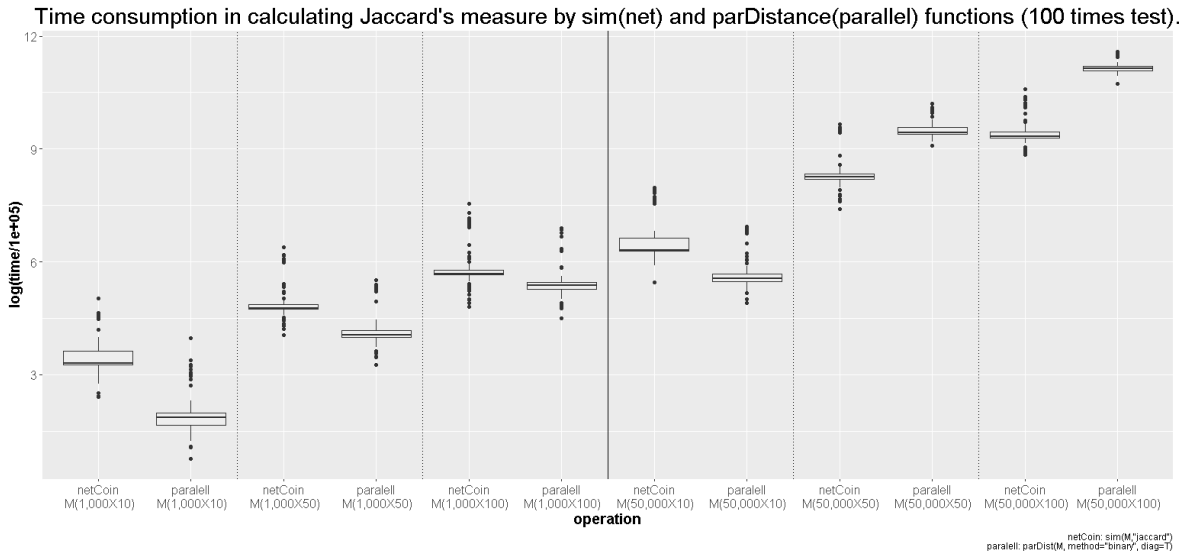

Figure 6: Performance of **netCoin** compared with **parallelDist**.

By means of routines from **igraph**, **netCoin** can reproduce different types of layouts and obtain communities with various algorithms, which facilitate the analysis and interpretation of coincidences. Data are then converted into D3 interactive graphs with controls enabling an interactive event analysis that can be shared with users online.

Acknowledgments

The work reported in this paper was supported by two grants to Modesto Escobar (CS-2013-49278-EXP and CSO2015-65094-P) from the State Program for R&D&i, whose funds comes from FEDER (European Union).

We are also grateful to the anonymous reviewers for comments on successive drafts of the paper, and to Carlos Prieto and David Barrios for their collaboration on **netCoin**.

References

Allaire JJ, Grandrud C, Rusell K, Yetman CJ (2015). ***networkD3**: **D3** JavaScript Network Graphs from R*. R package version 0.4, URL https://CRAN.R-project.org/package=networkD3.

Almende BV, Benoit T, Titouan R (2019). ***visNetwork**: Network Visualization Using **vis.js** Library*. R package version 2.0.9, URL https://CRAN.R-project.org/package=visNetwork.

Anderberg MR (1973). *Cluster Analysis for Applications*. Academic Press, New York.

Bastian M, Heymann S, Jacomy M (2009). "**Gephi**: An Open Source Software for Exploring and Manipulating Networks." In *International AAAI Conference on Weblogs and Social Media*.

Batagelj V, Mrvar A (1998). "**Pajek**-Program for Large Network Analysis." *Connections*, **21**(2), 47–58.

Bates D, Mächler M (2019). ***Matrix**: Sparse and Dense Matrix Classes and Methods*. R package version 1.2-18, URL https://CRAN.R-project.org/package=Matrix.

Baumgartner M (2009). "Inferring Causal Complexity." *Sociological Methods & Research*, **38**(1), 71–101. doi:10.1177/0049124109339369.

Baumgartner M, Thiem A (2015). "Identifying Complex Causal Dependencies in Configurational Data with Coincidence Analysis." *The R Journal*, **7**(1), 176–184. doi:10.32614/rj-2015-014.

Blei DM, Ng A, Jordan M (2003). "Latent Dirichlet Allocation." *Journal of Machine Learning Research*, **3**, 993–1022.

Blondel VD, Guillaume JL, Lefebvre E (2008). "Fast Unfolding of Communities In Large Networks." *Journal of Statistical Mechanisms: Theory and Experiment*, **2008**, P10008. doi:10.1088/1742-5468/2008/10/p10008.

Borgelt C (2012). "Frequent Item Set Mining." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, **2**(6), 437–456. doi:10.1002/widm.1074.

Bostock M, Ogievetsky V, Heer J (2011). "D^3 Data-Driven Documents." *IEEE Transactions on Visualization and Computer Graphics*, **17**(12), 2301–2309. doi:10.1109/TVCG.2011.185.

Butts CT (2008). "**network**: A Package for Managing Relational Data in R." *Journal of Statistical Software*, **24**(2), 1–36. doi:10.18637/jss.v024.i02.

Butts CT (2019). ***network**: Classes for Relational Data*. R package version 1.16.0, URL https://CRAN.R-project.org/package=network.

Carley K (1993). "Coding Choices for Textual Analysis; A Comparison of Content Analysis and Map Analysis." *Sociological Methodology*, **23**, 75–126. doi:10.2307/271007.

Chang J (2015). ***lda**: Collapsed Gibbs Sampling Methods for Topic Models*. R package version 1.4.2, URL http://CRAN.R-project.org/package=lda.

Chang W, Cheng J, Allaire JJ, Xie Y, McPherson J (2020). ***shiny**: Web Application Framework for R*. R package version 1.4.0.2, URL https://CRAN.R-project.org/package=shiny.

Connor EF, Simberloff D (1983). "Interspecific Competition and Species Co-Occurrence Patterns on Islands: Null Models and the Evaluation of Evidence." *Oikos*, **41**(3), 455–465. doi:10.2307/3544105.

Corman SR, Kuhn T, McPhee R, Dooley K (2002). "Studying Complex Discursive Systems: Centering Resonance Analysis of Communication." *Human Communication*, **28**(2), 157–206. doi:10.1111/j.1468-2958.2002.tb00802.x.

Cox TF, Cox MA (2001). *Multidimensional Scaling*. Chapman & Hall/CRC, Boca Raton.

Csardi G (2020). ***igraph*** *– The Network Analysis Package*. R package version 1.2.5, URL https://CRAN.R-project.org/package=igraph.

Csardi G, Nepusz T (2006). "The **igraph** Software Package for Complex Network Research." *InterJournal Complex Systems*, **1695**, 1–9.

Diaconis M, Mosteller F (1989). "Methods for Studying Coincidences." *Journal of the American Statistical Association*, **84**(408), 853–861. doi:10.1080/01621459.1989.10478847.

Diamond JM, Gilpin ME (1982). "Examination of the "Null" Model of Connor and Simberloff for Species Co-Occurrences on Islands." *Oecologia*, **52**(1), 64–74. doi:10.1007/bf00349013.

Dusa A, Thiem A (2020). ***QCA****: Qualitative Comparative Analysis*. R package version 3.7, URL https://CRAN.R-project.org/package=QCA.

Eckert A (2018). ***parallelDist****: Parallel Distance Matrix Computation Using Multiple Threads*. R package version 0.2.4, URL https://CRAN.R-project.org/package=parallelDist.

Epskamp S, Costantini G, Haslbeck J, Isvoranu A (2020). ***qgraph****: Graph Plotting Methods, Psychometric Data Visualization and Graphical Model Estimation*. R package version 1.6.5, URL https://CRAN.R-project.org/package=qgraph.

Epskamp S, Cramer AO, Waldorp LJ, Schmittmann VD, Borsboom D (2012). "**qgraph**: Network Visualization of Relationships in Psychometric Data." *Journal of Statistical Software*, **48**(4), 1–18. doi:10.18637/jss.v048.i04.

Escobar M (2015). "Studying Coincidences with Network Analysis and Other Multivariate Tools." *The Stata Journal*, **15**(4), 1118–1156. doi:10.1177/1536867x1501500410.

Escobar M, Barrios D, Prieto C, Martinez-Uribe L (2020). ***netCoin****: Interactive Analytic Networks*. R package version 1.1.25, URL https://CRAN.R-project.org/package=netCoin.

Everitt BS (2003). *The Cambridge Dictionary of Statistics*. Cambridge University Press, Cambridge.

Feinerer I (2019). *Introduction to the* ***tm*** *Package Text Mining in R*. R package version 0.7-7, URL https://CRAN.R-project.org/web/packages/tm/vignettes/tm.pdf.

Feinerer I, Hornik K (2019). ***tm****: Text Mining Package*. R package version 0.7-7, URL https://CRAN.R-project.org/package=tm.

Feinerer I, Hornik K, Meyer D (2008). "Text Mining Infraestructure in R." *Journal of Statistical Software*, **25**(5), 1–52. doi:10.18637/jss.v025.i05.

Finney DJ (1948). "The Fisher-Yates Test of Significance in 2×2 Contingency Tables." *Biometrika*, **35**(1–2), 145–156. doi:10.1093/biomet/35.1-2.145.

Fisher RA (1935). "The Logic of Inductive Inference." *Journal of the Royal Statistical Society*, **98**(1), 39–82. doi:10.2307/2342435.

Frick A, Ludwing A, Mehldau H (1995). "A Fast Adaptative Layout Algorithm for Undirected Graphs." In R Tamassia, IG Tollis (eds.), *Lecture Notes in Computer Science*, pp. 388–403. Springer-Verlag, Berlin. doi:10.1007/3-540-58950-3_393.

Fruchterman TMJ, Reingold EM (1991). "Graph Drawing by Force-Directed Placement." *Software: Practice and Experience*, **21**(11), 1129–1164. doi:10.1002/spe.4380211102.

Girvan M, Newman MEJ (2002). "Community Structure in Social and Biological Networks." *Proceedings of the National Academy of Sciences of the United States of America*, **99**(12), 7821–7828. doi:10.1073/pnas.122653799.

Good BH, de Montjove YA, Clauset A (2009). "The Performance of Modularity Maximization in Practical Contexts." *Physical Review E*, **81**, 046106. doi:10.1103/physreve.81.046106.

Gotelli NJ (2000). "Null Model Analysis of Species Co-Ocurrence Patterns." *Biology*, **81**(9), 2606–2621. doi:10.1890/0012-9658(2000)081[2606:nmaosc]2.0.co;2.

Gotelli NJ, Hart E, Ellison A (2015). ***EcoSimR**: Null Model Analysis for Ecological Data*. R package version 0.1.0, URL https://CRAN.R-project.org/package=EcoSimR.

Gower JC (1985). "Measures of Similarity, Dissimilarity, and Distance." In *Encyclopedia of Statistical Sciences*, volume 5. John Wiley & Sons, New York.

Grandrud C, Allaire JJ, Rusell K (2016). "**D3** JavaScript Network Graphs from R." URL http://christophergandrud.github.io/networkD3/.

Griffith DM, Veech JA, Marsh CJ (2016a). "**cooccur**: Probabilistic Species Co-Occurrence Analysis in R." *Journal of Statistical Software*, **69**(2), 1–17. doi:10.18637/jss.v069.c02.

Griffith DM, Veech JA, Marsh CJ (2016b). ***cooccur**: Probabilistic Species Co-Occurrence Analysis in R*. R package version 1.3, URL https://CRAN.R-project.org/package=cooccur.

Grimmer J, Stewart BM (2013). "Text as Data: The Promise and Pitfalls of Automatic Content Analysis Methods for Political Texts." *Political Analysis*, **21**(3), 267–297. doi:10.1093/pan/mps028.

Grosjean P (2014). ***tcltk2**: Tcl/Tk Additions*. R package version 1.2-11, URL https://CRAN.R-project.org/package=tcltk2.

Haberman SJ (1973). "The Analysis of Residuals in Cross-Classified Tables." *Biometrics*, **29**(1), 205–220. doi:10.2307/2529686.

Hahsler M, Grün B, Hornik K (2005). "**arules** – A Computational Environment for Mining Association Rules and Frequent Item Sets." *Journal of Statistical Software*, **14**(15), 1–25. doi:10.18637/jss.v014.i15.

Hamann U (1961). "Merkmalsbestand und Verwandtschaftsbeziehungen der Farinosae. Ein Beitrag zum System der Monokotyledonen." *Willdenowia*, **2**, 639–768.

Hubálek Z (1982). "Coefficients of Association and Similarity, Based on Binary (Presence-Absence) Data: An Evaluation." *Biological Reviews*, **57**(4), 669–689. doi:10.1111/j.1469-185x.1982.tb00376.x.

IBM Corporation (2017). *IBM SPSS Statistics 25*. IBM Corporation, Armonk. URL http://www.ibm.com/software/analytics/spss/.

Jaccard P (1901). "Distribution de la Flore Alpine dans le Bassin des Dranses et dans Quelques Règions Voisines." *Bulletin de la Societe Vaudoise des Sciences Naturelles*, **37**, 241–272. doi:10.5169/seals-266440.

Jurka T, Collingwood L, Boydstun AE, Grossman E, van Atteveldt W (2014). ***RTextTools**: Automatic Text Classification via Supervised Learning*. R package version 1.4.2, URL https://CRAN.R-project.org/src/contrib/Archive/RTextTools.

Kamada T, Kawai S (1989). "An Algorithm for Drawing General Undirected Graphs." *Information Processing Letters*, **31**(1), 7–15. doi:10.1016/0020-0190(89)90102-6.

Kembel SW, Cowan PD, Helmus MR, Cornwell WK, Morlon H, Ackerly DD, Blomberg SP, Webb CO (2010). "**Picante**: R Tools for Integrating Phylogenies and Ecology." *Bioinformatics*, **26**(11), 1463–1464. doi:10.1093/bioinformatics/btq166.

Kulczynski S (1927). "Die Pflanzenassoziationen der Pieninen." *Bulletin International de l'Academie Polonaise des Sciences et des Lettres, Classe des Sciences Mathematiques et Naturelles, B*, **Suppl. II**, 57–203.

Loiseau S, Vaudor L, Decorde M, Heiden S (2015). ***textometry**: Textual Data Analysis Package Used by the **TXM** Software*. R package version 0.1.4, URL https://CRAN.R-project.org/package=textometry.

Lucas C, Nielsen R, Roberts M, Stewart B, Storer A, Tingley D (2015). "Computer Assisted Text Analysis for Comparative Politics." *Political Analysis*, **23**(2), 254–277. doi:10.1093/pan/mpu019.

Lund K, Burgess C (1996). "Producing High-Dimensional Semantic Spaces from Lexical Co-Occurrence." *Behavior Research Methods, Instruments, & Computers*, **28**(2), 203–208. doi:10.3758/bf03204766.

Martin S, Brown WM, Klavans R, Boyack KW (2008). "DRL: Distributed Recursive (Graph) Layout." *Technical report*, Sandia National Laboratories.

Matsuo Y, Ishizuka M (2004). "Keyword Extraction from a Document Using Word Co-Occurrence Statistical Information." *International Journal of Artificial Intelligence Tools*, **13**(1), 157–169. doi:10.1142/s0218213004001466.

Meyer D, Buchta C (2019). ***proxy**: Distance and Similarity Measures*. R package version 0.4.23, URL https://CRAN.R-project.org/package=proxy.

Newman MEJ (2006). "Finding Community Structure in Networks Using the Eigenvectors of Matrices." *Physical Review E*, **74**, 1–22. doi:10.1103/physreve.74.036104.

Ochiai A (1957). "Zoogeographic Studies on the Soleoid Fishes Found in Japan and Its Neighbouring Regions." *Bulletin of the Japanese Society of Scientific Fisheries*, **22**(9), 526–530. doi:10.2331/suisan.22.526.

Oksanen J, Blenchet FG, Friendly M, Kindt R, Legendre P, McGlinn D, Minchin PR, O'Hara RB, Simpson GL, Solymos P, Stevens MHH, Wagner H (2019). ***vegan**: Community Ecology Package*. R package version 2.5-6, URL https://CRAN.R-project.org/package=vegan.

Padgett JF, Ansell CK (1993). "Robust Action and the Rise of the Medici, 1400–1434." *American Journal of Sociology*, **98**(6), 1259–1319. doi:10.1086/230190.

Pearson K (1900). "Mathematical Contributions to the Theory of Evolution. – VII. On the Correlation of Characters Not Quantitatively Measureable." *Philosophical Transactions of the Royal Society of London A*, **195**, 1–47. doi:10.1098/rsta.1900.0022.

Pons P, Latapy M (2006). "Computing Communities in Large Networks Using Random Walks." *Journal of Graph Algorithms and Applications*, **10**(2), 191–218.

Popping R (2000). *Computer-Assisted Text Analysis*. Sage Publications, London.

Popping R (2003). "Knowledge Graphs and Network Text Analysis." *Social Science Information*, **42**(1), 91–106. doi:10.1177/0539018403042001798.

Prieto C, Barrios D (2017). ***RJSplot**: Interactive Graphs with R*. R package version 2.5, URL https://CRAN.R-project.org/package=RJSplot.

Raghavan UN, Albert R, Kumara S (2007). "Near Linear Time Algorithm to Detect Community Structures in Large-Scale Networks." *Physical Review E*, **76**(3), 036106. doi:10.1103/physreve.76.036106.

Ragin C (1987). "The Comparative Method: Moving beyond Qualitative and Quantitative Methods." University of California, Berkeley.

Ragin CC (2000). *Fuzzy-Set Social Science*. University of Chicago Press, Chicago.

R Core Team (2020). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Reichardt J, Bornholdt S (2006). "Statistical Mechanics of Community Detection." *Physical Review E*, **74**, 016110. doi:10.1103/physreve.74.016110.

Reingold EM, Tilford JS (1981). "Tidier Drawings of Trees." *IEEE Transactions on Software Engineering*, **7**(2), 223–228. doi:10.1109/tse.1981.234519.

Roberts ME, Stewart BM, Tingley D (2019a). "**stm**: An R Package for Structural Topic Models." *Journal of Statistical Software*, **91**(2), 1–40. doi:10.18637/jss.v091.i02.

Roberts ME, Stewart BM, Tingley D (2019b). ***stm**: Estimation of the Structural Topic Model*. R package version 1.3.5, URL https://CRAN.R-project.org/package=stm.

Roberts ME, Tingley D, Lucas C, Leder-Luis J, Gadarian S, Alberrtson B, Rand D (2014). "Structural Topic Models for Open-Ended Survey Responses." *American Journal of Political Science*, **58**(4), 1064–1082. doi:10.1111/ajps.12103.

Robinson D, Silge J (2020). ***tidytext**: Text Mining Using **dplyr**, **ggplot2**, and Other Tidy Tools*. R package version 0.2.3, URL https://CRAN.R-project.org/package=tidytext.

Rogers DJ, Tanimoto TT (1960). "A Computer Program for Classifying Plants." *Science*, **132**(3434), 1115–1118. doi:10.1126/science.132.3434.1115.

Russell PF, Rao TR (1940). "On Habitat and Association of Species of Anopheline Larvae in South-Eastern Madras." *Journal of the Malaria Institute of India*, **3**, 153–178.

Sanderson J (2000). "Testing Ecological Patterns A Well-Known Algorithm from Computer Science Aids the Evaluation of Species Distributions." *American Scientist*, **88**(4), 332–339.

Schult DA, Swart P (2008). "Exploring Network Structure, Dynamics, and Function Using **NetworkX**." In *Proceedings of the 7th Python in Science Conference*.

Sneath PHA, Sokal RR (1962). "Numerical Taxonomy." *Nature*, **193**, 855–860. doi:10.1038/193855a0.

StataCorp (2019). *Stata Statistical Software: Release 16*. StataCorp LLC, College Station. URL http://www.stata.com/.

Strona G, Nappo D, Boccacci F, Fattorini S, San-Miguel-Ayanz J (2014). "A Fast and Unbiased Procedure to Randomize Ecological Binary Matrices with Fixed Row and Column Totals." *Nature Communications*, **5**(4114). doi:10.1038/ncomms5114.

Sugiyama K, Tagawa S, Mitsuhiko T (1981). "Methods for Visual Understanding of Hierarchical Systems Structure." *IEEE Transactions on Systems Man and Cybernetics*, **11**(2), 109–125. doi:10.1109/tsmc.1981.4308636.

Van Attenveld W (2008). *Semantic Network Analysis. Techiques for Extracting, Representing, and Quering Media Content*. Routledge, London.

Van Rossum G, *et al.* (2011). *Python Programming Language*. URL https://www.python.org/.

Veech JA (2013). "A Probabilistic Model for Analysing Species Co-Occurrence." *Global Ecology and Biogeography*, **22**(2), 252–260. doi:10.1111/j.1466-8238.2012.00789.x.

Wakita K, Tsurumi T (2007). "Finding Community Structure in Mega-Scale Social Networks." arXiv:cs/0702048 [cs.CY], URL https://arxiv.org/abs/cs/0702048.

Wasserman S, Faust K (1994). *Social Network Analysis: Methods and Applications*, volume 8. Cambridge University Press, Cambridge.

Wickham H, Miller E (2019). ***haven**: Import and Export SPSS, Stata and SAS Files*. R package version 2.2.0, URL https://CRAN.R-project.org/package=haven.

Young L, Soroka S (2012). "Affective News: The Automated Coding of Sentiment In Political Texts." *Political Communication*, **29**(2), 205–231. doi:10.1080/10584609.2012.671234.

Yule GU (1900). "On the Association of Attributes in Statistics: With Illustrations from the Material of the Childhood Society." *Philosophical Transactions of the Royal Society of London A*, **194**, 257–319. doi:10.1098/rsta.1900.0019.

Zhang J (2016). ***spaa**: SPecies Association Analysis*. R package version 0.2.2, URL https://CRAN.R-project.org/package=spaa.

Affiliation:

Modesto Escobar
Department of Sociology and Communication
Faculty of Social Sciences
University of Salamanca
37071 Salamanca, Spain
E-mail: modesto@usal.es
URL: http://sociocav.usal.es/web/en/miembros/sociologia/escobar-mercado/

Luis Martinez-Uribe
Doctoral Program in Social Sciences
University of Salamanca

Journal of Statistical Software http://www.jstatsoft.org/
published by the Foundation for Open Access Statistics http://www.foastat.org/
May 2020, Volume 93, Issue 11 *Submitted:* 2017-06-15
doi:10.18637/jss.v093.i11 *Accepted:* 2019-12-04