Evaluation and Program Planning 60 (2017) 284–292

Contents lists available at ScienceDirect

Evaluation and Program Planning

journal homepage: www.elsevier.com/locate/evalprogplan

R-CMap—An open-source software for concept mapping

Haim Bar, PhD.[a,*], Lucas Mentch, PhD.[b]

[a] Department of Statistics, University of Connecticut, United States
[b] University of Pittsburgh, Dept. of Statistics, United States

ARTICLE INFO

Article history:
Received 26 July 2016
Accepted 22 August 2016
Available online 28 August 2016

Keywords:
Concept mapping
Open-source
The R programming language

ABSTRACT

Planning and evaluating projects often involves input from many stakeholders. Fusing and organizing many different ideas, opinions, and interpretations into a coherent and acceptable plan or project evaluation is challenging. This is especially true when seeking contributions from a large number of participants, especially when not all can participate in group discussions, or when some prefer to contribute their perspectives anonymously. One of the major breakthroughs in the area of evaluation and program planning has been the use of graphical tools to represent the brainstorming process. This provides a quantitative framework for organizing ideas and general concepts into simple-to-interpret graphs. We developed a new, open-source concept mapping software called R-CMap, which is implemented in R. This software provides a graphical user interface to guide users through the analytical process of concept mapping. The R-CMap software allows users to generate a variety of plots, including cluster maps, point rating and cluster rating maps, as well as pattern matching and go-zone plots. Additionally, R-CMap is capable of generating detailed reports that contain useful statistical summaries of the data. The plots and reports can be embedded in Microsoft Office tools such as Word and PowerPoint, where users may manually adjust various plot and table features to achieve the best visual results in their presentations and official reports. The graphical user interface of R-CMap allows users to define cluster names, change the number of clusters, select rating variables for relevant plots, and importantly, select subsets of respondents by demographic criteria. The latter is particularly useful to project managers in order to identify different patterns of preferences by subpopulations. R-CMap is user-friendly, and does not require any programming experience. However, proficient R users can add to its functionality by directly accessing built-in functions in R and sharing new features with the concept mapping community.

1. Introduction

“Well begun is half done”[1]; is the essence of project planning and the key to the final outcome of the project. However, planning complex projects usually involves input from many sources, including future consumers, managers, engineers, etc. Assembling and analyzing input from diverse sources with varying levels of expertise and different priorities, and converting these inputs into project objectives and detailed plans of execution is a major challenge. Planners require tools and methods to assess which goals are perceived as important and feasible, and how they relate to each other. To this end, the group concept mapping approach provides the visualization and analytical tools with which planners can:

1. identify specific project goals,
2. determine the required resources and expertise,
3. assign priorities, and
4. define objective measures of success.

Group concept mapping helps planners achieve these goals by employing a six-step process (Trochim & McLinden, 2017). In this article we focus on two of the six steps, namely, (i) the multivariate statistical analysis, which provides the mathematical foundations for the inferential process, and (ii) the generation and interpretation of the concept maps, which facilitate the analysis by summarizing input from a large number of stakeholders in graphical form. These two steps are critical to understanding the composite thinking among a diverse group of stakeholders with varying levels of expertise, knowledge, responsibility, and

[*] Corresponding author.
E-mail address: haim.bar@uconn.edu (L. Mentch).

[1] Attributed to Aristotle.

http://dx.doi.org/10.1016/j.evalprogplan.2016.08.018
0149-7189/© 2016 Elsevier Ltd. All rights reserved.

interest, and to translating the input into a prioritized plan of action.

We introduce a new, freely available, open source concept mapping tool, called R-CMap, which not only implements the traditional multivariate analysis and graphical representations, but also introduces new analytical and graphical tools for concept mapping. Furthermore, it allows users who are familiar with the statistical programming language R to add functionality and features. Based on similar experiences in other fields (e.g. bioinformatics) we anticipate that the release of an open source concept mapping software will contribute to accelerated advances in the fields of planning and evaluation, as more researchers will be able to contribute and incorporate new methods and visualization tools. In the following section we very briefly describe the programming language R and some of its main features and capabilities. We then discuss some implementation details and provide examples of its statistical and graphical capabilities, and conclude with a discussion.

2. A brief introduction to R

The R programming language (R Core Team, 2016) is a powerful programming tool used extensively throughout the scientific community to perform statistical analysis, computing, graphing, and data mining that is freely available under the terms of the GNU General Public License as published by the Free Software Foundation, on multiple platforms, including Windows, MacOS, and linux. R contains a wide range of statistical tools, including linear regression, hypothesis testing, categorical data analysis, time series, and clustering, just to name a few. It is constantly evolving, with new features and tools becoming available frequently. R users also benefit from its collaborative nature, since they can download contributed packages that extend the core (but very extensive) functionality of the language, or make their own implementations available to the public as packages, typically distributed via the Comprehensive R Archive Network (http://www.cran.r-project.org). Installing and updating contribute packages is very simple even to the novice user, via the install.packages() and update.packages() commands.

R is an implementation of the S programming language (Becker & Chambers, 1984), which was first designed and developed at Bell Labs in 1975–1976. The first version of R became available in 1997, and became officially part of the GNU project (www.gnu.org). The GNU project was initiated by Richard Stallman from MIT in 1983, with the goal of developing free software. R is free of charge and can be distributed under the GNU General Public License, but the word "free" in "free software" actually refers more broadly to the user's freedom to use the software, adapt it to their needs, and share it by redistributing the original, as well as adapted versions. To read more about the free-software philosophy and how it differs from "open-source", see (Stallman, 2014).

R programs are "scripted", which means that the user does not need to perform additional steps to run the code. In particular, in contrast to some other languages (like C and Java), one does not have to compile or link the code, and R programs are usually entirely portable from one operating system to another, meaning that in most cases, an R program which was created on one computer will run on any other computer without needing any modifications. Another advantage of scripted languages, such as R, is that one can run individual lines or commands, without running an entire program. This is very useful for debugging, performing simple calculations, and creating customized functions.

Most practitioners use R in a "command line" mode, as opposed to a "graphical mode". Thus, people who are used to Excel or Visual Basic, or other menu-driven tools, may find it less intuitive initially. However, many tools, such as RStudio (RStudio, 2015), offer user-friendly features like viewing current values of variables and data structures and viewing and manipulating plots to assist less technical users. Two very useful recent additions to the RStudio development environment include "Shiny" (Chang, Cheng, Allaire, Xie, McPherson, 2016) which allows developers to deploy R programs as web applications, and "knitr" (Xie, 2016) which allows users to integrate text with R code and easily generate publication-ready documents. The current version of R-CMap uses the shiny package in a local-host mode, which means that when one starts the program, "Shiny" starts a web-server on the user's machine. We will soon introduce a version of our software which uses the remote-host mode, which means that the web server runs on a remote machine which is accessible through the internet. The remote-host version allows users to use R-CMap without having to install R or any additional packages on their computer.

In addition to the tools like RStudio and the documentation provided by authors of thousands of free packages, users can also find countless tutorials, online documentation sources, active discussion groups, and books. This contributes to the fast rate in which R is adopted by different sectors in both academia and industry. Though this flexibility and functionality are available, it is worth stressing again that our R-CMap package can be executed via a GUI, eliminating the need for technical programming skills from users.

3. R-CMap—statistical and graphical capabilities

3.1. Foundational improvements and extensions

Implementing our concept mapping software in R allows users across a variety of platforms to have open and free access to well-known concept mapping functionalities. Another major advantage to using an open-source language such as R is that we may incorporate a wide variety of statistical and graphical functions already available in order to extend the current concept mapping methodology to new situations, and develop improvements at each stage of the process to help users better understand the data. Some of the extensions and improvements described below are available in the current version while others require more extensive refinement and will be released in future updates.

3.2. New options for multidimensional scaling

Throughout this paper, we discuss our procedures in terms of a classic framework in which a number of participants are given several statements and asked to sort them into piles as they see fit. The sorted piles are represented as incidence matrices, one for each participant, where cell (i,j) contains 1 if the participant put statements i and j in the same pile and 0 if the statements were placed in different piles. The incidence matrices are added, and their sum is denoted by S. It is then normalized by dividing by the number of participants, to construct a similarity matrix, denoted by S^*. The normalization step insures that if two statements co-occur in the same pile for every participant, then the similarity score is 1. On the other end of the spectrum, a similarity score of 0 represents two statements that were not put in the same pile by any participant. Equivalently, one can consider the derived distance matrix, $D = I - S^*$, where I is the identity matrix, and both I and S^* are $k \times k$ matrices, and k is the number of statements. Thus, column i in the distance matrix contains all the pairwise distances between statement i and all the other statements. These pairwise distances are not sufficient to determine the overall configuration of all k statements in a k-dimensional space. Furthermore, it is impossible to visualize a k-dimensional space. Therefore, the next step in the concept mapping process involves transforming the

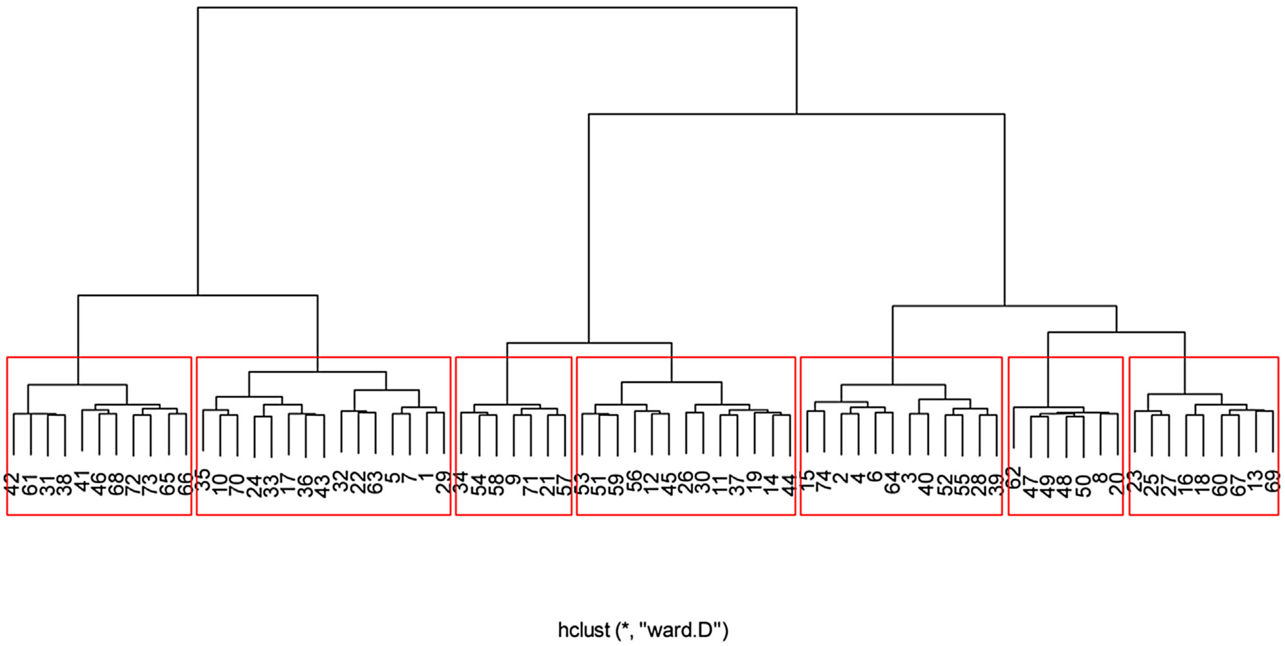


Fig. 1. A dendrogram to determine the number of clusters.

high-dimensional data into a two-dimensional space, while trying to preserve the pairwise distances as much as possible.

This transformation is done using the multidimensional scaling (MDS) algorithm (Kruskal, 1964). The extent to which the original relative distances are preserved in the two-dimensional space is measured by the “stress” statistic. Informally speaking, the more the MDS algorithm successfully preserved pairwise distances, the lower the stress. Then, one proceeds by clustering statements in the two-dimensional space. Generally, a configuration with lower stress will yield a more reliable clustering since the distances in the two-dimensional space are better approximations of the original distances in the high dimensional space.

Note that the definition of the matrix D uses a specific notion of distance, but in multivariate analysis there are multiple ways to define distance between two statements. We list some distance measures below, but first point out that in many cases it is not clear which one is most appropriate for the data.

3.3. Possible distance measures—quantitative data

Let s_{i} and s_{j} be the i-th and j-th columns of the matrix S. Let d_{ij} be a distance measure between statements s_{i} and s_{j}. We consider the following formulas for d_{ij} (Cox & Cox, 2001):

$$d_{ij} = \frac{|s_{i1} - s_{j1}|}{s_{i1} + s_{j1}} + \ldots + \frac{|s_{ik} - s_{jk}|}{s_{ik} + s_{jk}} \qquad \text{Canberra :}$$

$$d_{ij} = \sqrt[p]{|s_{i1} - s_{j1}|^p + \ldots + |s_{ik} - s_{jk}|^p} \qquad \text{Minkowskimetric :}$$

$$d_{ij} = \sqrt{\left(s_{i1} - s_{j1}\right)^2 + \ldots + \left(s_{ik} - s_{jk}\right)^2} \qquad \text{Euclideandistance :}$$

$$d_{ij} = |s_{i1} - s_{j1}| + \ldots + |s_{ik} - s_{jk}| \qquad \text{Cityblockmetric :}$$

$$d_{ij} = max\{|s_{ir} - s_{jr}| : r = 1, \ldots, k\} \qquad \text{Maximum :}$$

Note that the Minkowski metric can be seen as a generalization of the latter three distances, with $p = 2$, $p = 1$ and $p = \infty$, respectively. The Canberra distance is intended for non-negative values s_{ik} (as in the case in concept mapping).

Two generalizations of the Euclidean matrix are the weighted-Euclidean, and the Mahalanobis distance. In addition, we can define a distance based on the correlation between s_{i} and s_{j}, or based on the angle between the vectors s_{i} and s_{j}. One may also use the binary method, where s_{i} and s_{j} are considered as binary vectors (non-zero values are converted to 1).

The user can try different definitions of distance and select the one which minimizes the stress. For example, using a concept mapping data set, we obtain the following stress values:

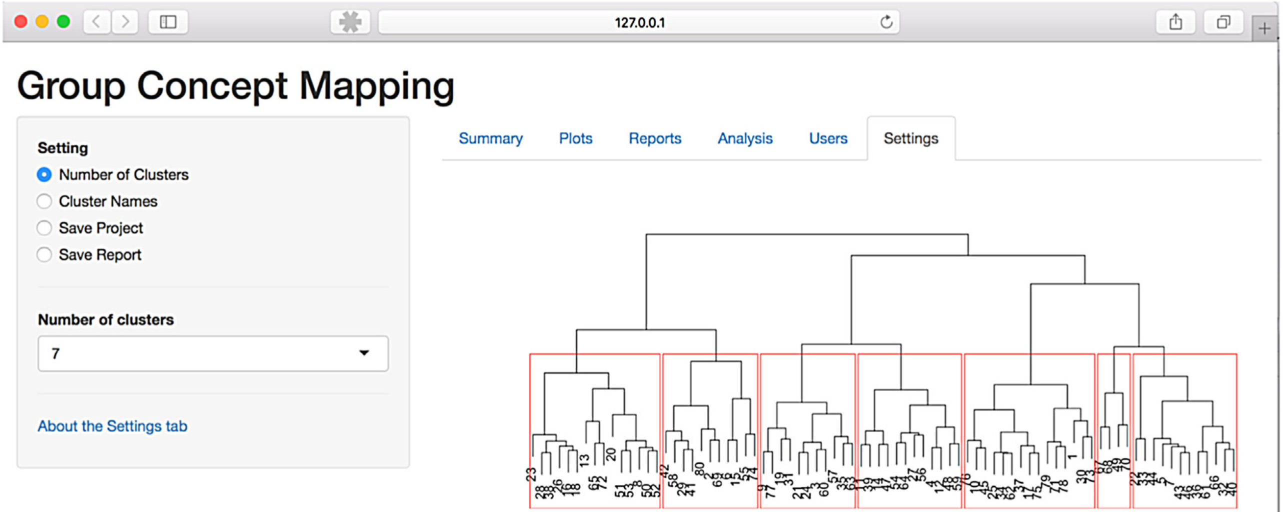
- Maximum: 29.68
- Minkowski metric (p = 3): 20.92
- Euclidean: 20.43
- City block: 19.56
- Canberra: 16.69

Thus, the choice of the distance metric can lead to significant improvement in stress, and consequently to improved clustering.

4. New options for hierarchical clustering

4.1. Choosing the clustering method and the number of clusters

After performing the MDS, the user is provided a 2-dimensional representation of the relative distances between statements: statements that were often grouped together by participants appear closer together in the two-dimensional plot, and those that seldom coexist in one pile are placed farther apart. To further simplify the data and increase the interpretability of the results, the next step is to form clusters of statements that capture more general themes, or concepts. To do that, the user can choose the clustering method and the number of clusters. Currently, in other concept mapping tools, only one hierarchical clustering option is provided (Ward's method, Ward, 1963), and users are provided little guidance other than visual intuition as to how many clusters to choose.



Group Concept Mapping

Setting

● Number of Clusters
○ Cluster Names
○ Save Project
○ Save Report

Number of clusters

7

About the Settings tab

Fig. 2. Selecting the number of clusters.

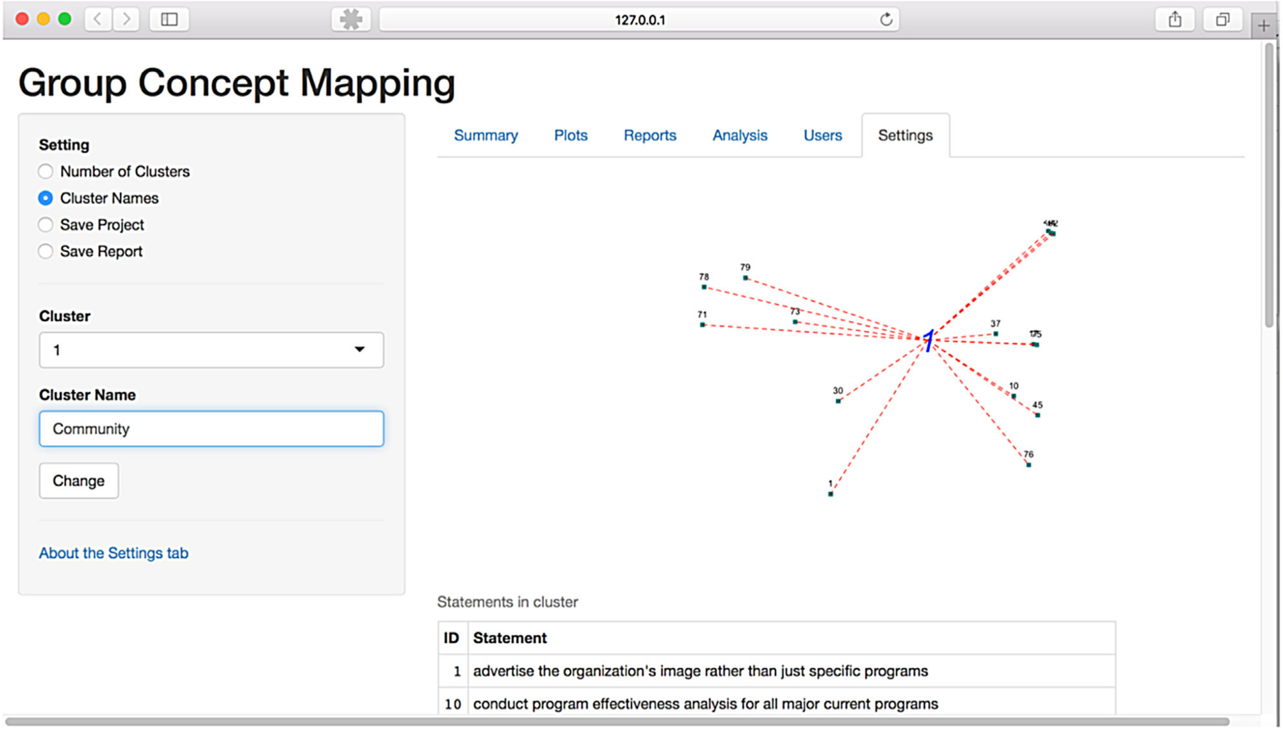
Using R allows us to offer a number of clustering methods to the users. There are multiple packages that implement clustering algorithms, and for a user who is comfortable with R, it is straightforward to select one of them instead of the default algorithm (hclust, Murtagh, 1985). Using our software, it is easy to choose between clustering methods. Generally, the determination of clusters is done by finding an optimal configuration, where “optimal” can be defined in a number of ways, usually by optimizing some objective function. For example, one may try to maximize the distance between clusters, or minimize the total variance within clusters. Here are a few examples:

4.1.1. Complete linkage

the distance between two clusters is the maximum possible distance between a point in one cluster and a point in the other cluster. In the iterative clustering algorithm we start with each point being its own cluster, and in each iteration the two closest clusters are combined. The process ends when all points are in one cluster. The user has to choose the number of clusters, and the iterative process ends when that number of clusters is reached. Mathematically, the distance between pairs of clusters, A and B, is defined as: $max\{d(a,b): a{\in}A,\ b{\in}B\}$.

4.1.2. Single linkage

Also known as “nearest neighbor clustering”. The process is similar to the complete linkage, except that the distance between two clusters is determined by the closest pair of points, such that each point in the pair belongs to a different cluster. Mathematically, we define the distance between cluster A and cluster B as $min\{d(a,b): a{\in}A,\ b{\in}B\}$.



Group Concept Mapping

Setting

○ Number of Clusters
● Cluster Names
○ Save Project
○ Save Report

Cluster

1

Cluster Name

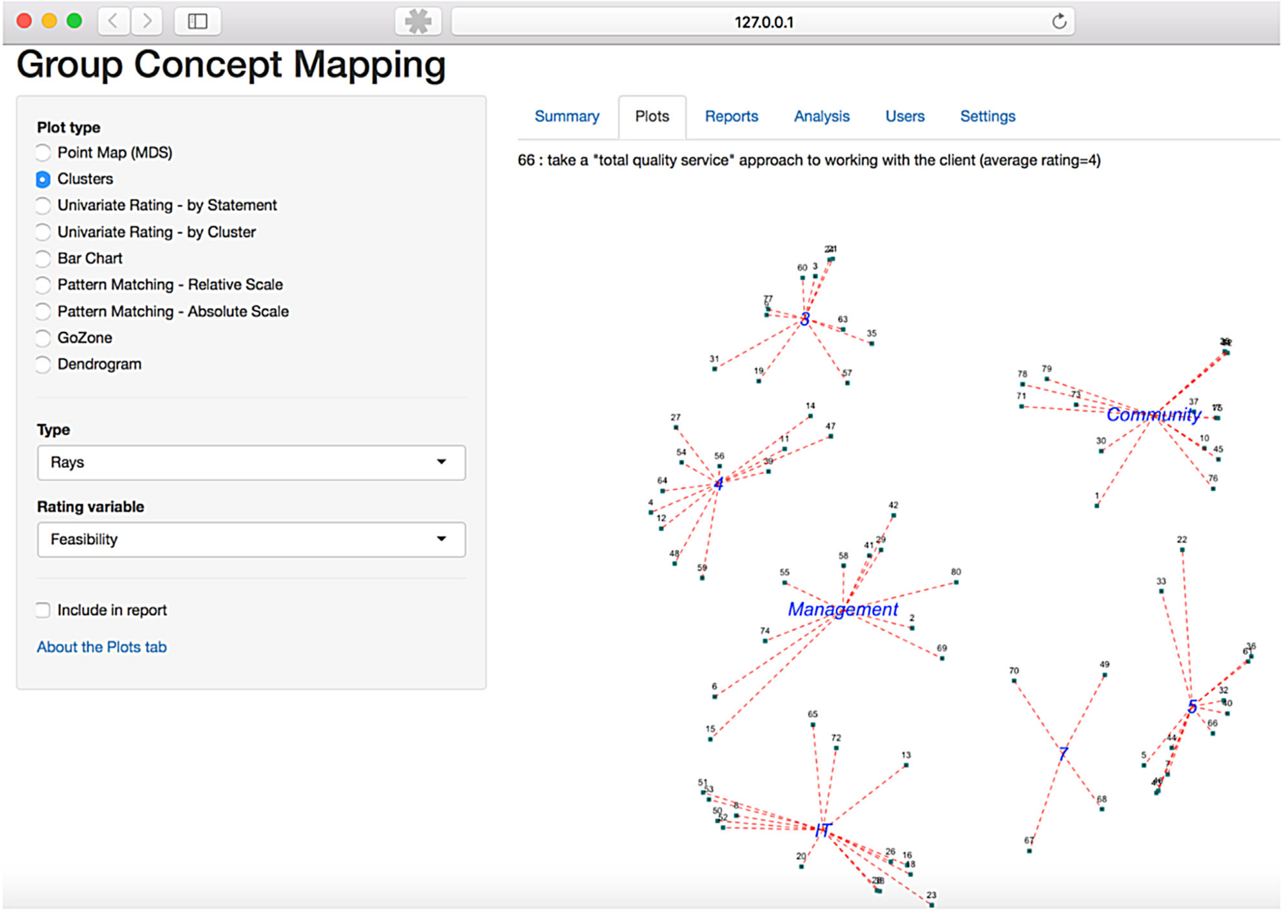
Community

About the Settings tab

Statements in cluster

ID	Statement
1	advertise the organization's image rather than just specific programs
10	conduct program effectiveness analysis for all major current programs

Fig. 3. Choosing a cluster label.



Group Concept Mapping

Plot type
○ Point Map (MDS)
● Clusters
○ Univariate Rating - by Statement
○ Univariate Rating - by Cluster
○ Bar Chart
○ Pattern Matching - Relative Scale
○ Pattern Matching - Absolute Scale
○ GoZone
○ Dendrogram

Type
Rays

Rating variable
Feasibility

☐ Include in report

About the Plots tab

66 : take a "total quality service" approach to working with the client (average rating=4)

Fig. 4. A cluster map (rays).

4.1.3. *Ward's method*

Here, the objective is to minimize the variance within clusters. The iterative process starts with singleton clusters, and in each iteration the algorithm finds the two clusters which, when merged, lead to the smallest increase in total within-cluster variance. Variance within a cluster is measured as the sum of squared distances from points in a cluster to its center. Thus, Ward's method achieves good results when the true clusters have a spherical shape.

Other available methods included in the hclust function are average, median, and centroid. Alternative clustering methods like k-means clustering and various Bayesian methods are also available in R and can be easily incorporated into our software. Though the statistical community does not consider any one particular method to be universally better than the others, some procedures come with desirable asymptotic properties and can be more useful in various contexts. We do not expand on such considerations any further, as it is beyond the scope of this paper. For more information on cluster analysis, see Everitt, Landau, Leese, Stahl, 2011.

To help the user select the number of clusters, we introduce an option to view a dendrogram – a tree representation of the clustering process, in which the length of stems represents the distance between two merged clusters. Readers unfamiliar with dendrograms can refer to (Everitt et al., 2011) for further details. For example, in Fig. 1 we have seven clusters that are represented by the red rectangles. Each rectangle contains the statements that were clustered together. Generally, the chosen number of clusters should not cause the upper side of the rectangles to cut any short stems, as these stems represent closely related subsets, as opposed to easily distinguishable clusters. On the other hand, the number of clusters should not be too small, either; if the upper side of a rectangle cuts long stems then the rectangle contains two or more subsets which could form distinct clusters. When Ward's method is used, the length of a stem corresponds to the additional within-cluster variability as a result of merging two smaller clusters. Our software can be used to show the numerical value of total within-cluster variability for different settings (as a function of the number of clusters) in order to further assist the user in choosing the "optimal" configuration. Of course, the user can decide on a different number of clusters when the alternative results in more intuitive concepts.

4.2. *Rating-based clustering, and clustering subjects*

In addition to clustering statements based on their co-occurrences in user pile-sorting, it can also be useful to see how statements are clustered based on similarity in their ratings. Given k rating variables, we can associate with each statement a k-dimensional vector consisting of the average ratings provided by the participants. We then proceed by computing the matrix of distances between the vectors of statement ratings, running MDS (if $k > 2$), and performing the hierarchical clustering. When $k = 2$, this process yields clusters of statements in the go-zone plot. When

$k > 2$, we get a clusters in the (two dimensional) scaled version of the multivariate response space, which can be seen as a multivariate extension of the go-zone plots.

Furthermore, it is also possible, and indeed, interesting, to cluster the *subjects* themselves based on their sorting or rating. Using demographic data, it is then possible to see whether some clusters correspond to certain demographic variables, such as rank in the organization, age, sex, etc.

5. Clustering—user interface options

After selecting the number of clusters, the next step is usually to try determine whether the statements in a particular cluster can be represented by a broader description, or concept, that describes the statements in the cluster. Fig. 2 shows a screenshot of the user interface to select the number of clusters based on the dendrogram. Fig. 3 shows the interface for setting the cluster names. The user can choose a cluster number from the drop down menu on the left, and set the name based on the list of statements in the cluster (only two are shown in this screenshot), and suggested names (not shown here) based on pile labels. In some applications of pile-sorting, participants are told to provide a descriptive label for each pile. These labels appear after the list of statements along with a weight which our software computes for each label. These labels should help the user choose an appropriate name for the cluster. In this example, the user chose the name “Community”. Fig. 4 shows a “ray-map” which consists of seven clusters.

6. Demographic and ratings-based inference

The current version of our software includes several inferential tools such as the ladder-style ratings plots, reports of overall ratings and ratings by demographic, tools to examine the data by demographic, and go-zones. While these tools are very helpful in discovering patterns in the way users group and rate the statements, the construction of our software in R makes a variety of new and helpful inferential tools readily available.

One of the simplest but perhaps most helpful tools we provide is the ability to test for rating equality across clusters. That is, given two or more clusters, it is straightforward to perform a formal hypothesis testing procedure and obtain *p*-values for determining whether the average ratings differ across clusters. Specifically, using R to build the concept mapping tool allows us to easily integrate capabilities such as *t*-test for comparing pairs of clusters, or more generally, analysis of variance (ANOVA) to test whether mean ratings differ across several clusters. Furthermore, this is easily extended to analysis of covariance (ANCOVA) or a regression model, where, in addition to including the clusters as a predictor, the user can also include demographic variables to explain variability in some rating variable. When multiple rating variables are available, multivariate regression methods can be used. For example, one can perform principal component analysis and use the first principal components to create the familiar univariate or bivariate plots (e.g. cluster rating map, go-zone). Nonparametric procedures (e.g. the Kruskal Wallis test (Kruskal & Wallis, 1952)) can also be easily incorporated into the software, in order to handle cases where the rating variables do not fit a normal model or when some clusters contain very few observations.

The following tables show an excerpt from a report that our software generates. The first table shows the mean feasibility rating for all statements in each cluster, and the standard deviation. In this example, the feasibility rating associated with the “Community” cluster is the lowest (2.72). The second table contains the ANOVA result, which tests whether the feasibility is statistically the same in all seven clusters. Since the *p*-value is very small, we conclude that not all the clusters have the same feasibility rating. Adding more tests is straightforward. For example, one can add contrasts to test more specific hypotheses.

Summary by Cluster

Cluster ID	Feasibility	S D
Community	2.72	0.39
Management	3.825	0.36
IT	3.271	0.4
4	3.614	0.46
5	3.846	0.51
6	3.078	0.38
7	3.291	0.65

Analysis of Variance

DF	Sum Sq.	Mean Sq.	F value	p-value
6	13.088	2.18129	10.201	5.3792e-08
67	14.326	0.21383		

In addition to testing whether mean ratings are equal across clusters, the user may be interested in testing whether the standard deviations of ratings differs across clusters. This can be used to assess the level of controversy or disagreement regarding a particular group of statements within certain clusters. For example, suppose that political party affiliation is one of the demographic variables in the data (democrat and republican) and that the participating individuals are given statements consisting of potential political issues or programs. Both groups might naturally group together statements related to, for example, forms of social welfare programs, but democrats and republicans would certainly rate these statements differently if asked, “How likely are you to vote in favor of this legislation?”

We also provide more intuitive, finer-scale plots to help users better understand underlying connotations associated with the statements. For more background on these go-zone plots, we refer the reader to (Kane & Trochim, 2007). Perhaps the most straightforward example of this is providing a go-zone plot for each cluster. The “global-level” go-zone consisting of all statements from all clusters is helpful for understanding which particular statements have particularly high or low ratings, but does not contain the information regarding how the statements were clustered. To include this information, we provide two additional types of go-zone plots. The first plot still contains all statements, but the statements will be color-coded according to cluster. The second contains only those statements included in a given cluster, or set of clusters, selected by the user. The horizontal and vertical lines in the go-zone corresponding to the mean rating on each scale will remain calculated with respect to the entire collection of statements so that users can see how ratings from within a particular cluster compare to the overall ratings. For example, Fig. 5 shows go-zone plots for two clusters that have been named “Client” (left) and “Benefits” (right). In addition to showing the mean rating on each variable, in this case, importance and feasibility, we also include lines for ±SD in each dimension.

We also provide bar plots of different ratings by cluster, and it is also possible to create these plots for different subsets of the sorters. For example, Fig. 6 shows the average ratings (±SD) for feasibility and importance in a six-cluster configuration, for a subset of users based on the size of organization (the barplots in this example summarize the ratings from small organizations). The average importance rating is nearly uniform across clusters, but in terms of feasibility three clusters are considered more feasible (Management, Benefits, and Communication).

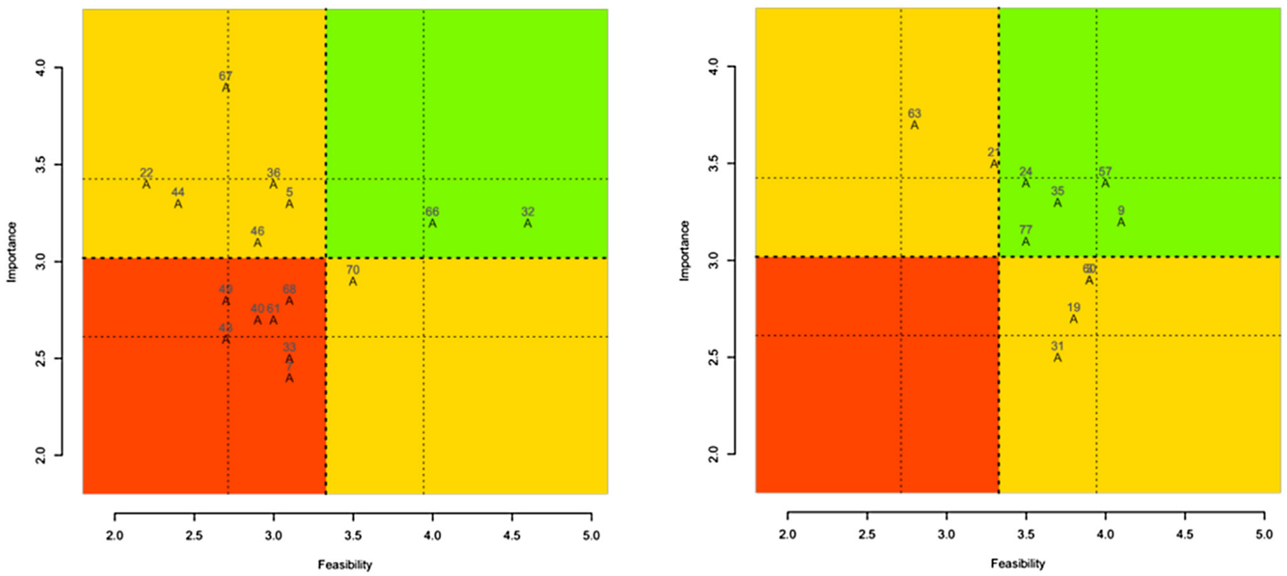


Fig. 5. Go-zone plots by cluster. The middle lines represent the overall means of the two ratings (e.g. feasibility and importance). The plots also contain the lines for ±SD in each dimension. Note that in the Benefits cluster (right), most statements are rated highly according to both rating scales while in the Client cluster (left), many statements received a lower than average feasibility rating.

Finally, it is important to keep in mind the crucial yet often overlooked aspect of any statistical analysis of defining the appropriate population on which the analysis is to apply. Users implementing a concept mapping approach who are able to include all relevant parties in the sample (e.g. business executives, hospital administrators) may not need to perform the statistical procedures outlined above as, in this case, differences between demographic responses represent the true, population-level differences. Nonetheless, these observed differences would apply only to the particular employees (participants) at the particular time of the experiment. Treating the participants as a true sample and performing some type of statistical analysis would allow users

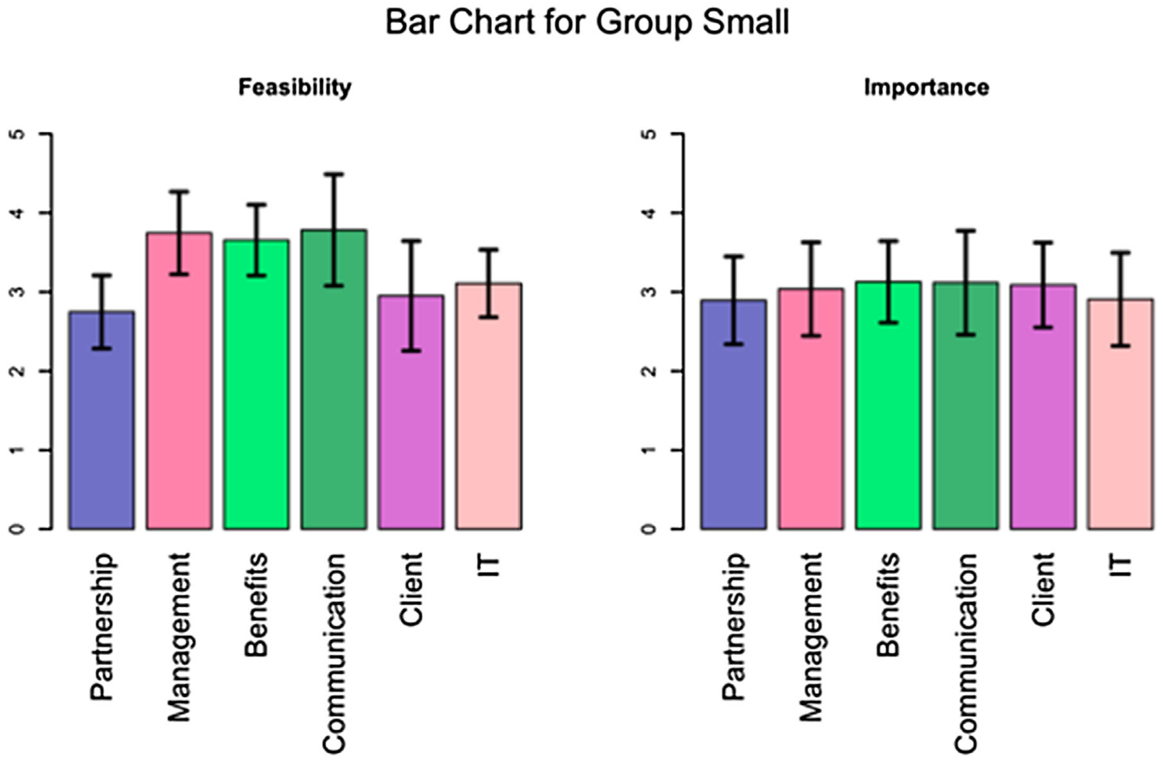


Fig. 6. Barplots of two rating variables for a subset of sorters (from a group called "Small").

to generalize the results to a broader context and take into account variability introduced by, for example, hiring an additional employee.

7. Model selection

We previously mentioned the dendrogram as an effective graphical tool to assess which configuration, in terms of the number of clusters, is most appropriate for the sorted statement data. Here we elaborate on this topic and mention additional methods to check how many clusters best represent the data:

- Total sum of squared errors within clusters (SSW) – compute the squared distance from each statement to the center of its cluster. “Center” can be defined as the average of all the points in the cluster, but other definitions are also possible. For example, one can use the coordinates of the most central statement in the cluster. When using Ward’s method, one chooses the number of clusters by observing at which point the increase in SSW becomes too large.
- Maximize the ratio SSB/SSW, where SSB is the between-cluster sum of squares and SSW is the within-cluster sum of squares. We calculate SSW as before, and we calculate the sum of squared distances between cluster centers, and the overall average of the 2-dimensional configuration of the statements. This ratio is similar in spirit to the familiar F test in an ANOVA framework. The larger the ratio SSB/SSW, the better the separation of the clusters. (To avoid the possibility that each statement will be in its own cluster, we define SSB/SSW = 0 if SSW = 0). This provides a formal statistical test for determining the optimal number of clusters, under certain assumptions. Specifically, this test requires that the two-dimensional representation of the statements follows a mixture of bivariate normal distributions. This is closely related to Fisher’s Linear Discriminant Analysis method (Fisher, 1936; Mardia, Kent, Bibby, 1979). Note that the “ray-map” (as in Fig. 2) can be used to visually assess the assumption of bivariate normal distribution of the scaled data. If the assumption does not seem reasonable, one can easily implement non-parametric alternatives for the SSB/SSW test. For example, resampling or permutation approaches (e.g. the bootstrap, Efron, 1979) may be used. The details of these tests are beyond the scope of this paper.

In addition to selecting the most appropriate number of clusters, we may also want to fit a model to the rating variables. In this case, we can use likelihood-based model selection methods. For example, one can use AIC (Akaike Information Criterion, Akaike, 1974) or BIC (Bayes Information Criterion, Schwarz, 1978) to select the best model for fitting a rating variable. In particular, one can fit the model using different number of clusters, and check which configuration minimizes the information criterion, or determine which demographic variable is associated with the rating variable. Similarly, the sequential F-test criterion, or Mallows’ Cp criterion can also be used (Mallows, 1973) for selecting the appropriate model.

When more than one rating variable is available, one can formally compare clusters by their bivariate distribution. Visually, this amounts to testing whether the cluster-specific go-zones plots are significantly different. One way to do it analytically is to consider the go-zone plots as 2×2 contingency tables, and test if two such tables are statistically different.

8. Missing data, and data collection

Before generating the distance matrix and performing the analysis it is important to check the completeness of the data. Each user must put each statement in exactly one pile. Our software provides a warning message when a participant has either failed to group a statement, or when a particular statement has been assigned to multiple groups by the same participant. We also provide the user with information regarding which statement(s) are missing or duplicated. Experimenters must then amend the data file in order to perform the usual concept mapping techniques.

However, in certain situations it may be reasonable for users not to sort some statements. For example, if users feel that they do not have all the necessary information, it may be better to avoid sorting a statement as opposed to sorting that statement arbitrarily. In principle, this type of incomplete data should not affect the analysis. The dissimilarity matrix just has to be normalized differently, where, instead of dividing the counts by the number of participants, we use a different weight for each statement, and these weights depend on the number of participants who did sort each statement. The rating averages have to be adjusted accordingly.

If some rating or demographic data is missing, then we can either use a complete case analysis, where we discard any incomplete response, or we can use one of a number of imputation methods which are readily available in R (for example, multiple imputation (Rubin, 1987), or propensity score matching algorithm (Rosenbaum & Rubin, 1983)). We anticipate allowing this imputation option in a future release so that the concept mapping analysis may still be performed in the presence of missing data.

To avoid the missing data problem as much as possible, we plan to develop a web-based data entry tool that will allow users to easily perform the sorting step, enter their demographic data, and their provide their ratings. Such a tool will allow us to check the completeness of the data interactively, and prompt the participant to correct any problems. Furthermore, such a tool will allow users of concept mapping tools to include a much larger number of participants in the process, since the manual effort involved in the data collection step will be significantly reduced.

9. Conclusion

This paper introduced new software for performing concept mapping analysis. We demonstrated new features and capabilities including the ability to perform statistical tests to compare ratings across clusters and alternative graphical representations. This extends easily to more sophisticated linear models, where the user can control for other covariates, such as demographic variables. We also explained how our software can be used to generalize known tools, such as the go-zones and discussed how the software can help the user reach more informed decisions about the selected number of clusters, and more generally how to choose a model which provides the best fit for the data. We plan to introduce other features and capabilities which may be useful for project planning and evaluation, beyond those that were mentioned in this article. For example, one may consider performing network analysis on the statements. This will include introducing notions such as “centrality” or “connectivity” of nodes (statements, but also possibly stakeholders). This can be seen as a generalization of the clustering step. When statements are clustered, we simply assume that all statements in one cluster are connected (via a common concept), and we do not attempt to assign strengths to connections between pairs of statements. In reality, some statements are more closely related than others, and they may also be related (perhaps to a lesser degree) to statement in other clusters.

We wrote the software in R in order to provide users with a very wide range of functionality, including inferential tools and new graphical options. The choice to use R was also motivated by the philosophy of “free software”, which advocates that source code should be freely available to the users who are then free to make modifications, add features, and share with the community of other users. We believe that our software will be applicable in several disciplines, beyond project evaluation and planning. In particular, we envision that social scientists and anthropologists will find it useful as analogous techniques are common in these fields. We hope that once the software is released an expert group will be established in order to design the next version of the software, and reach a consensus on which features and statistical procedure should be added. We believe that a cross disciplinary collaboration will be very productive and will lead to many important improvements and innovations in the area of evaluation and program planning.

The program is available for download from the first author's web-site (http://haim-bar.uconn.edu/) or by request via email (haim.bar@uconn.edu).

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, *19*(6), 716–723.

Becker, R. A., & Chambers, J. M. (1984). *S: an interactive environment for data analysis and graphics*. Pacific Grove, CA, USA: Wadsworth & Brooks/Cole.

Chang, Winston, Cheng, Joe, Allaire, J. J., Xie, Yihui, & McPherson, Jonathan (2016). *Shiny: web application framework for R. r package version 0.13.2.*. . https://CRAN.R-project.org/package=shiny.

Cox, T. F., & Cox, M. A. A. (2001). *Multidimensional scaling*. Chapman and Hall.

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics*, *7*(1), 1–26.

Everitt, B. S., Landau, S., Leese, M., & Stahl, D. (2011). *Cluster analysis*, 5th ed. Chichester, UK: John Wiley & Sons, Ltd..

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, *7*(2), 179–188.

Kane, M., & Trochim, W. M. K. (2007). *Concept mapping for planning and evaluation. Applied social research methods series, Vol 50*, .

Kruskal, & Wallis (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, *47*(260), 583–621.

Kruskal, J. B. (1964). Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, *29*, 1–27.

Mallows, C. L. (1973). Some comments on CP. *Technometrics*, *15*(4), 661–675.

Mardia, K. V., Kent, J. T., & Bibby, J. M. (1979). Multivariate analysis. *Probability and mathematical statistics series*. Academic Press.

Murtagh, F. (1985). *Multidimensional clustering algorithms, in COMPSTAT lectures 4*. Wuerzburg: Physica-Verlag.

R Core Team (2016). *R: a language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. http://www.R-project.org/.

RStudio Team (2015). *RStudio: integrated development for R*. Boston, MA: RStudio, Inc.. http://www.rstudio.com/.

Rosenbaum, R. Paul, & Rubin, B. Donald (1983). The central role of the propensity score in observational studies for causal effects. *Biometrika*, *70*(1), 41–55.

Rubin, D. B. (1987). *Multiple imputation for nonresponse in surveys*. New York: Wiley & Sons.

Schwarz, E. Gideon (1978). Estimating the dimension of a model. *Annals of Statistics*, *6*(2), 461–464.

Stallman, Richard (2014). *Why open source misses the point of free software*. . https://www.gnu.org/philosophy/open-source-misses-the-point.html.

Trochim, M. William, & McLinden, Daniel (2017). Introduction to a special issue on concept mapping. *Evaluation and Program Planning*, *60*, 166–175.

Ward, J. H. Jr. (1963). Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, *58*, 236–244.

Yihui, Xie (2016). *knitr: a general-purpose package for dynamic report generation in R. R package version 1.13.* .

Haim Bar received his Ph.D. in statistics at Cornell University in 2012 (Dissertation Topic: "Parallel Testing, and Variable Selection – a Mixture-Model Approach with Applications in Biostatistics". Advisors: James Booth (Chair), Martin T. Wells, Robert L. Strawderman). He received his M.Sc. in statistics in 2010 (Cornell University) and an M.Sc. in computer science in 2002 (Yale University). He received his bachelor degree in mathematics (Cum Laude) in 1993, at the Hebrew University in Jerusalem. His professional interests include statistical modeling, shrinkage estimation, high throughput applications in biology (e.g., genomics, brain imaging), Bayesian statistics, variable selection, and machine learning. From 1995 to 1997, he was with Motorola, Israel, as a computer programmer in the Wireless Access Systems Division. From 1997 until 2003 he worked for MicroPatent, LLC, where he held the position of Director of Software Development. In 2003 he moved to Ithaca, NY, and worked as a Principal Scientist at ATC-NY.

Lucas Mentch obtained his Ph.D. in Statistics from Cornell University in 2015 and is currently an Assistant Professor in the Department of Statistics at the University of Pittsburgh. During the 2015–2016 academic year, Lucas was on leave participating in the program on statistics in forensic science at the Statistics and Applied Mathematical Sciences Institute (SAMSI).