Comput Math Organ Theory (2018) 24:149–168
https://doi.org/10.1007/s10588-017-9252-6

MANUSCRIPT

Blockmap: an interactive visualization tool for big-data networks

Terrill L. Frantz[1]

Published online: 5 April 2017

Abstract This article describes the Blockmap, which is a mechanism for displaying and exploring network datasets. The data are presented in a squarified-mosaic form, which is well-suited for visual display on a computer or phone screen. The relational data are dimension-reduced and structured for interactive, hierarchical exploration. The Blockmap applies a combination of treemap and heatmap display schemes specifically to the analysis of large network datasets. The Blockmap offers the analyst a way to explore underlying node-level data, at the full-network level, according to shared characteristics of the constituent nodes. It offers a technique for exploring nodesets—collections of network nodes—which have been classified according to a user-defined set of rules or discriminative algorithms. Typically, nodes can be classified according to their common attributes or a stratification of their ego-level network measures, but means can be extended. Using a Blockmap, an analyst can profile a network according to the meaningful characteristics exhibited by the mosaic; this technique also offers theorists a platform for developing a methodological and analytic framework for characterizing and analyzing network data. Production versions of Blockmap technology are presently hosted in client- and web-based software and is available freely in *ORA-LITE.

Keywords Network analysis · Large datasets · Data visualization · Treemap · Heatmap · Software

✉ Terrill L. Frantz
terrill@phbs.pku.edu.cn

[1] Peking University HSBC Business School, Shenzhen, China

1 General introduction

In the present day, the abundance of relational data for study and analysis has exceeded the human capacity to comprehend unaided. Inexpensive electronic storage of data and Internet-enabled collection of relational data have provided great opportunity for the social sciences. The endless data and processing technology allows researchers to delve into intricate details of social behavior and relationships at the micro-, meso-, and macro-level of analysis. While machine-based capacity such as computing technology and advances in machine learning have been advancing, innate human-based capacities to process such complex data and at such size remains bounded.

Forever, humans likely will continue to rely on our eyes as a sensor for making sense of our environment and as an input channel for analyzing information such as network data. In recent years, there has been much innovation towards the visual presentation of raw data in modified form that greatly aids in the analysis of data. Building on long-existing and commonplace methods such as bar charts and pie charts, creative schemes such as waterflow charts and bubble charts are now offered and easy to produce via Microsoft, Google and other ubiquitous computer software platforms. Advancing computer and display technologies have enthusiastically embraced these ingenious visualization techniques nearly making the traditional methods almost passé.

However, specific to relational network data, the traditional node-link diagram remains the mainstay. Indeed, computing hardware has kept up with the computational demands of the increasing size of datasets as millions of nodes and never-ending links can be presented on a computer screen either as pixel-based or as vector drawings. However, our human capacity to cognize such detailed images is severely limited by our biological and cognitive capacities, even when aided by the indispensable zoom feature.

Visual data exploration is essential activity in any research domain. Exploratory data analysis (EDA) (Tukey 1977) of *social network* data is no exception (Newman and Leicht 2007; de Nooy et al. 2005). EDA is a common first-step in social network analysis and will undoubtedly remain so in the future (Freeman 2000a). Visual representation of source data is also an important aid in presenting analysis findings to others (Freeman 2000b). Given the present-day amplified size of relational datasets, social network analysts have been calling for alternatives to the traditional node-link network visualization (Viégas and Donath 2004). Analysts can no longer rely exclusively on traditional visualization methods.

As Fig. 1 illustrates, node-link diagrams remain a powerful tool for understanding simple relational data sets, such as a 10-node, 20% density undirected network. Conversely, as Fig. 2 illustrates, a node-link visualization of just a 100-node, 20% density undirected network provides little aid in evolving our understanding of the underlying data. The limitation of the node-link diagram is readily apparent, indeed. While the node-link diagram maintains its immense utility, its usefulness is limited to a decreasing portion of the network datasets being studied presently. Better ways of visually exploring network data are necessary.

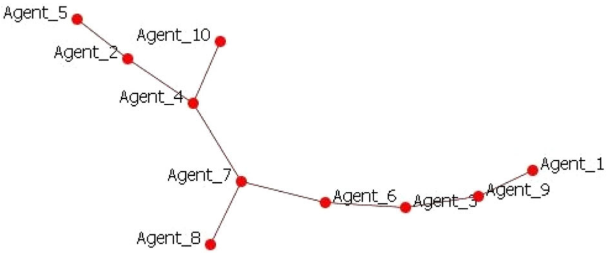


Fig. 1 Node-link diagram of 10-node, 20% density undirected network dataset

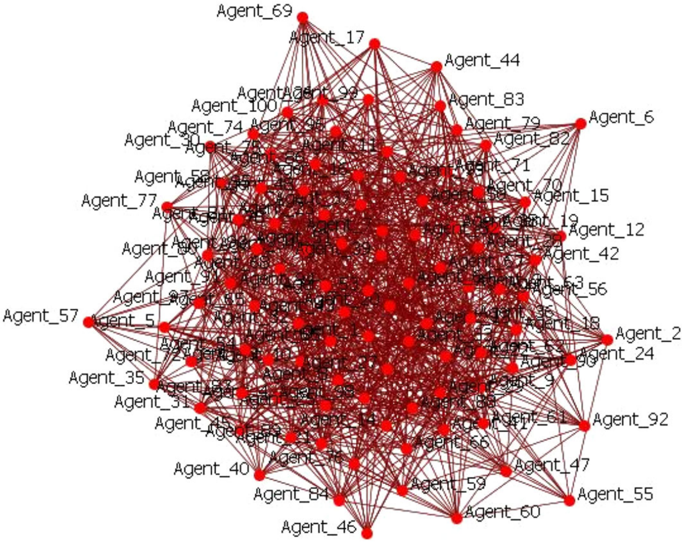


Fig. 2 Node-link diagram of 100-node, 20% density undirected network dataset

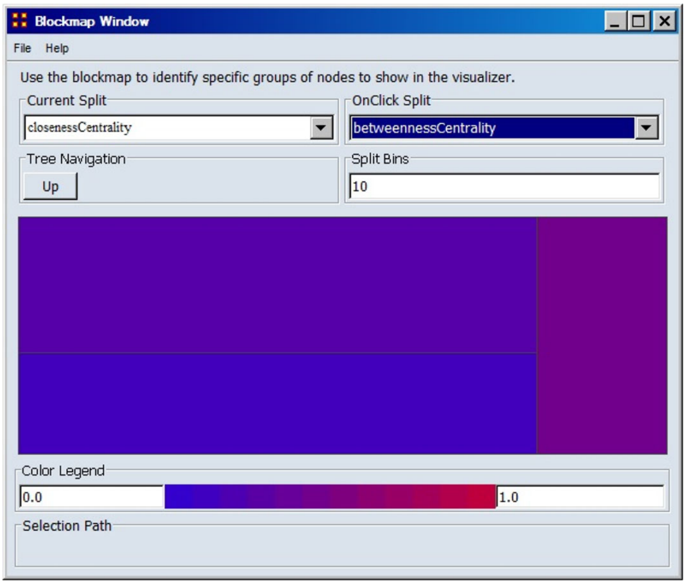
Over the past two decades, along with the ubiquitous computing power, researchers have recognized the limitations of node-link diagrams (Keller et al. 2006; Saket et al. 2014; Ware and Bobrow 2005) and have been exploring alternative ideas with some success (Frantz and Carley 2005a, b; Frisch et al. 2009; Sathiyanarayanan and Burlutskiy 2015a, b). These creative ideas are quite powerful, but some suffer either from a narrow field of situations where the solution is applicable, or the solution ultimately suffers from the similar data-size limitations of the node-link diagram (Holten 2006).

2 Introducing the blockmap

The blockmap is an innovation that addresses the dataset size limitations inherent to the node-link diagram; blockmaps are data-volume insensitive. A representation of data via a blockmap does not change in its physical dimensions or resolution according to the volume of data it is presenting. Thus, a key characteristic of the blockmap technology is that its usefulness is not inhibited by the volume of underlying network data, either by the number of nodes or the number of ties. The two-dimensional area or resolution of a block map of a 10-node, 100-node, or one-million-node is the same, as is the amount of space taken on a display medium—usually a computer or phone display or on printed hardcopy. Figure 3 illustrates a block map for the same data underlying that illustrated in Fig. 1; Fig. 4 is similarly paired with Fig. 2.

As Figs. 3 and 4 both show, a blockmap is a construction in a squarified-or rectangular-mosaic form. This four-sided shape is well-suited for conventional hardcopy print, and electronic display on a computer or phone screen; a feature that can be appreciated by comparing against the relative space inefficiency of a pie chart, for example. There is nothing necessary inherent in the rectangular presentation of the blockmap. Some creatively applications could easily be imaged where the shape can be anything from a circle to a cloud, even to a human silhouette. Greater explanation of the blockmap, how to create it, and how to interpret it is provided in a following section.

Another productive feature of a blockmap is that it is well-suited for interactive engagement by a human—or machine/software for that matter. This situates the blockmap for being a tool that goes beyond mere data presentation but one that can be used for exploration and thus as an aid to an EDA exercise. By using mouse-click technology, the analyst can drill-down or –up, as in navigating within a hierarchical tree (Blanch and Lecolinet 2006), to explore various partitions of the data according



Use the blockmap to identify specific groups of nodes to show in the visualizer.

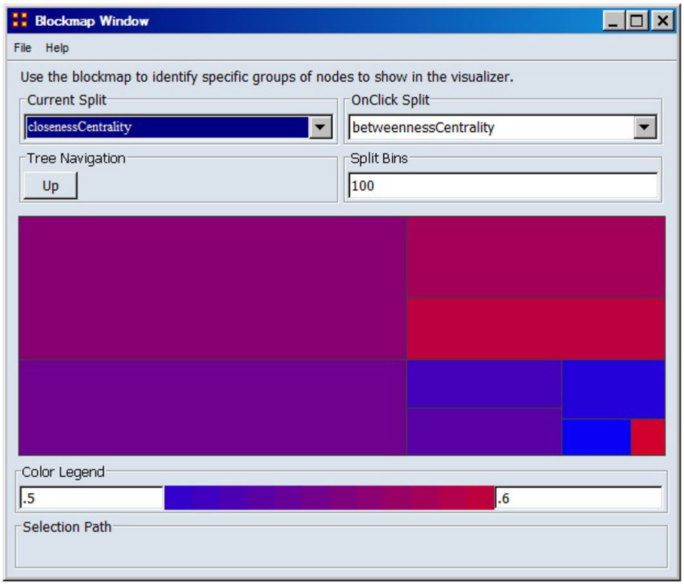
Current Split: closenessCentrality

OnClick Split: betweennessCentrality

Split Bins: 10

0.0 1.0

Fig. 3 Blockmap of 10-node, 20% density undirected network dataset, segmented by closeness centrality measure



Use the blockmap to identify specific groups of nodes to show in the visualizer.

Current Split: closenessCentrality

OnClick Split: betweennessCentrality

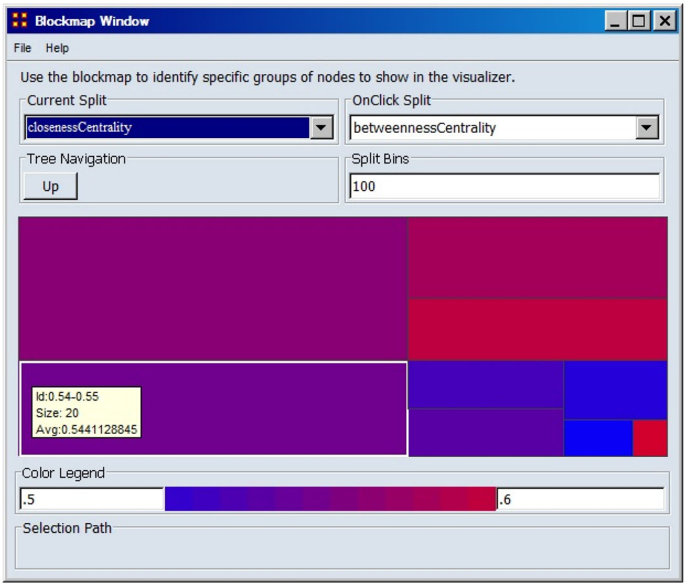
Tree Navigation

Split Bins: 100

Color Legend: .5 – .6

Selection Path

Fig. 4 Blockmap of 100-node, 20% density undirected network dataset, segmented by closeness centrality measure



Use the blockmap to identify specific groups of nodes to show in the visualizer.

Current Split: closenessCentrality

OnClick Split: betweennessCentrality

Tree Navigation

Split Bins: 100

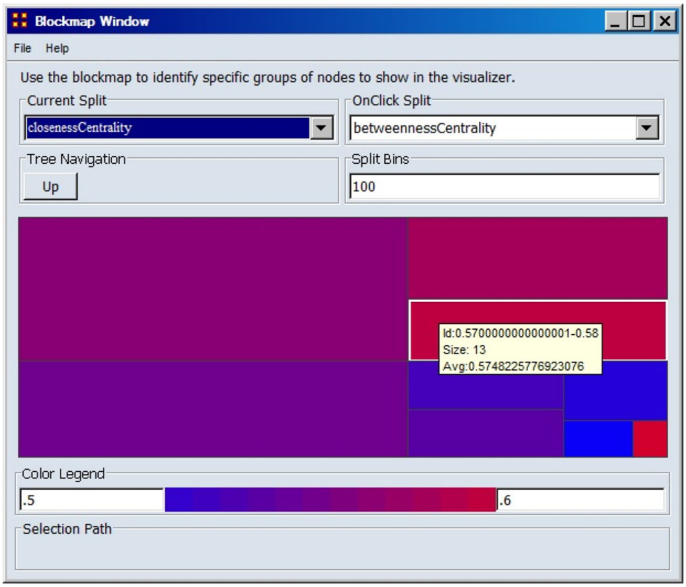
Id:0.54-0.55
Size: 20
Avg:0.5441128845

Color Legend: .5 – .6

Selection Path

Fig. 5 Blockmap of 100-node, 20% density undirected network dataset, mouse-over on a partition representing 20 nodes

to designated categorization rules or discriminative algorithm. Figures 5 and 6 illustrate how a mouse-over can provide additional information about the underlying portioned group.



Use the blockmap to identify specific groups of nodes to show in the visualizer.

Current Split: closenessCentrality

OnClick Split: betweennessCentrality

Tree Navigation

Split Bins: 100

Id:0.5700000000000001-0.58
Size: 13
Avg:0.5748225776923076

Color Legend: .5 .6

Selection Path

Fig. 6 Blockmap of 100-node, 20% density undirected network dataset, mouse-over on a partition representing 13 nodes

Of course, to accompany the scale-free benefit of the blockmap, there is some loss of information—relative to the node-link diagram—that the analyst must be cognizant of. A blockmap loses dyadic tie information, such as the presence of a tie, or not, as well as the directionality and the weight of the relationship. However, if such information is essential to analysis, such information could be captured in the construction of the blockmap; the information would be presented as an attribute transformed from the original form.

Blockmap technology is derived from an amalgamation of treemap and heatmap display schemes. These underlying component parts are the additive building blocks of the blockmap technology. A thorough understanding of their features can be applied directly to blockmap technology; these component parts are described in the next section.

3 Underlying technologies

The Blockmap is conceived and constructed as a combination of two pre-existing visualization technologies, a treemap and a heatmap, which is applied specifically to relational, network data. To understand the features of a blockmap it is well-suited to understand the underlying treemap and heatmap technologies. First, the treemap is presented, followed by an introduction to the heatmap.

3.1 Treemap

Treemap technology was inspired by a drive towards presenting data in a two-dimensional rectangular space (Johnson and Shneiderman 1991). The technology was initially applied to the mapping of computer disk-storage (Bederson and Shneiderman 2003). The name *treemap* originates from transforming a computer-file directory tree into a planar space-filling map (Shneiderman and Plaisant 1998).

In a treemap, hierarchical data appears as a mosaic image, which is illustrated in Fig. 7. Figure 7 (Babaria 2001) illuminates the hierarchal aspects of the treemap for population data for individual states of the United States of America. The hierarchy is rooted with USA as the country shown at the top of the graphic, and supported by geographic areas, e.g., West, Rockies, etc., shown immediately below, and finally the component states for each region beneath.

Figures 8, 9, and 10 (Rouse 2017) collectively illustrate the treemap hierarchical underpinning coupled with the block-area sizing feature. Each tile in the mosaic represents a specific datum, such that the two-dimensional area of each individual tile reflects the relative value, or size, of the underlying datum. This is akin to various characteristics and features of a pie chart, though presented in rectangular form; such similarity specifically includes the proportionality aspects of the pie-slice area relative to the whole. A computational algorithm (Bruls et al. 2000; Cesarano et al. 2016) computes the relative area for each tile and transforms the 4-sided shape into length and width dimensions (Kolatch and Weinstein 2001) so that all of the tiles fit perfectly into a fixed-dimension rectangular, mosaic image.

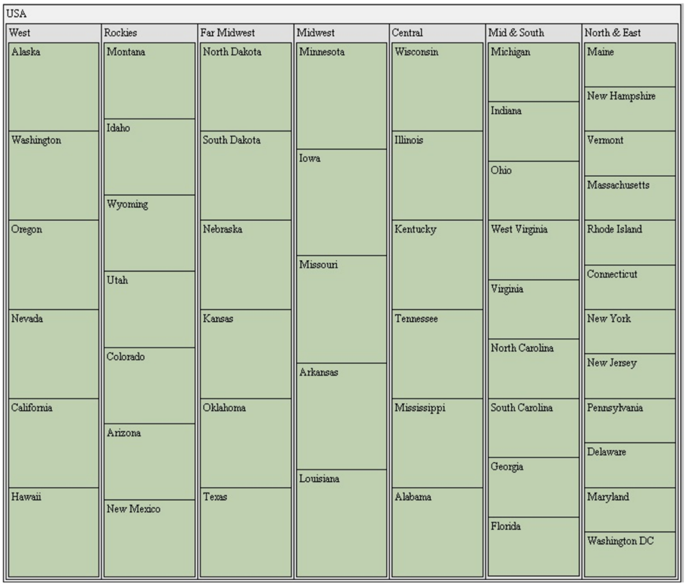


Fig. 7 Example treemap—USA states according to geographic region (Babaria 2001)

Fig. 8 Example tree structure (Rouse 2017)

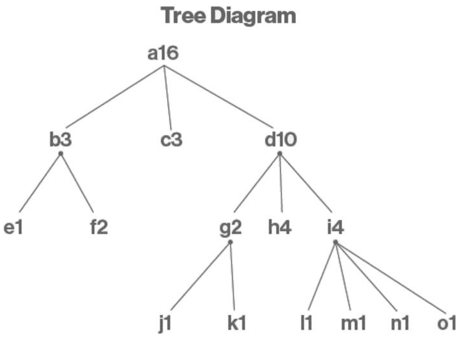


Fig. 9 Venn diagram representation of tree structure shown in Fig. 8

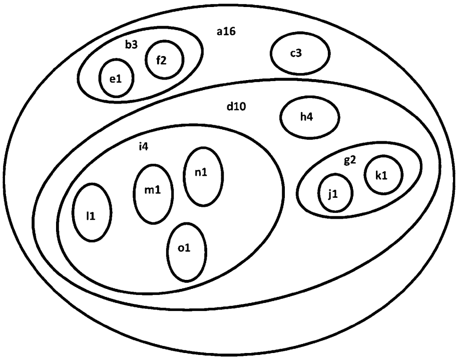
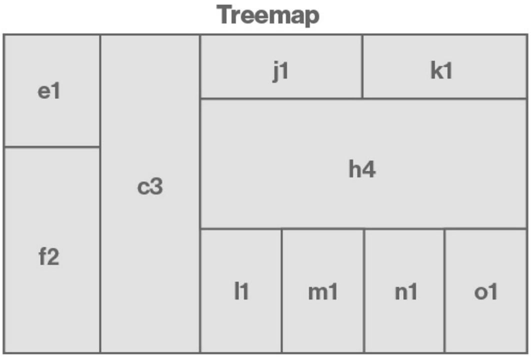


Fig. 10 Treemap representation of tree structure shown in Fig. 8 (Rouse 2017)



A potent feature of treemap technology is its interactivity. The treemap is typically displayed on a computer screen, rather as a static, or printed, display. An analyst can explore the hierarchy of the data with simultaneously being resented with the quantitative aspects. Computer mouse-overs are often a feature as well; these allow for additional information to be presented to the user. These features of the treemap allow for visual exploration of large datasets without a degradation of resolution of taking additional display area—a feature adopted by blockmaps.

The most significant feature of the treemap is its ability to represent large datasets, which has been demonstrated with as many one million items (Fekete and Plaisant, 2002a, b). This single feature makes treemaps distinctively relevant to big-data network researchers, as discussed in the Introduction section above.

3.2 Heatmap

A heatmap shows a data, typically in a mosaic form, where the coloring of each tile is indicative of an underlying quantitative value for the data represented by the tile. The special color, sometimes grey shape, is gradient-based so that the viewer can visually identify areas that are high or low in value, and any shade between. The origins of the heatmap can be traced to 1873 (Wilkinson and Friendly 2009) and has recently become somewhat a common method for displaying multitudes of data in a readily recognizable form. It has wide application in practice and research (Chu et al. 2012; Friendly, 1994; Lin et al. 2013). The heatmap has extensively been adopted by the financial markets for displaying stock-related information as Fig. 11 (4-traders.com 2017) illustrates. This heatmap applies treemap technology to the S&P 500 stock index and classified according to the business sector. The size of each tile is according to the market capitalization of the specific stock component.

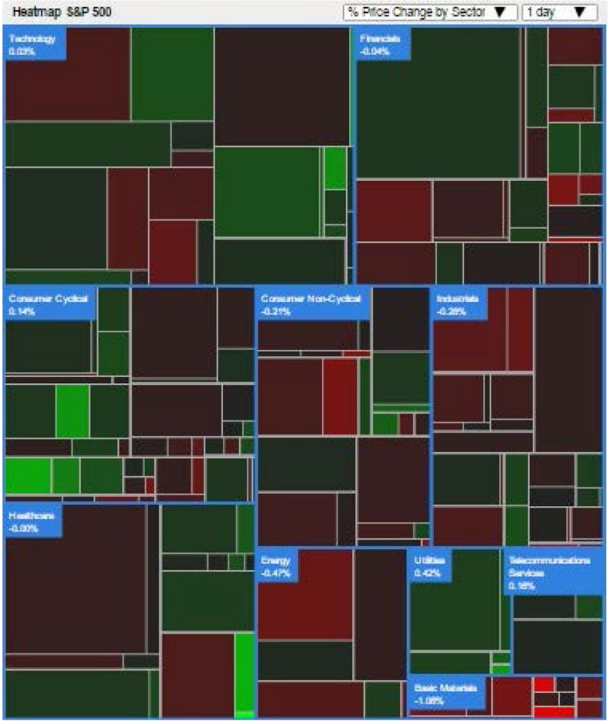


Fig. 11 Example heatmap—Components of the S&P 500 Index according to day’s change (4-traders.com 2017)

The green color shaping specifies a rise in the price for the day and its color shading and intensity indicates the degree of the rise, while the red color-base indicates a decrease in the price for the day.

4 Blockmap and the big-data network

While the time-honored, node-link diagram remains indispensable to network analysis, the blockmap ought not be blindly considered a replacement for the node-link diagram. The blockmap, however, may be an indispensable replacement for the node-link diagram for exploratory data analysis of big-data networks. The blockmap is well suited for overcoming size-caused limitations of the graph-link diagram. Blockmap design characteristics include: efficiency of computer screen display, the ability to effectively present large networks, and interactive capability. These features position the blockmap impeccably for big-data network exploration (Frantz and Carley 2005a, b).

The underpinning of the blockmap is the ease in which nodes in a network dataset can be clustered into meaningful collections, or subgroups, and displayed as such. The analysts often discriminates individual nodes into subgroups when exploring the data at a level above the individual or dyad. Creation of subgroups can be accomplished by exploring the prospects in a hit-or-miss manner or by way of computational or statistical methods. Regardless of how the nodes are actually clustered into groups, the blockmap handles the processing regardless of how the nodes are discriminated into subgroups.

When examining the results of the grouping process, the analyst may review the membership of the subgroups and make comparisons across and between the groups by noticing the area-size of the tile and the color of it. A tile in a blockmap typically represents a collection of nodes, but conceivably a tiles could, and meaningfully might, represent a single node, just the same.

Blockmaps have the capability to display both hierarchal (Corominas-Murtra et al. 2013; Holten 2006; Kules et al. 2003) and categorical information simultaneously and be worked interactively. Users stipulate node clusters by specifying categories based on the values of the specific attribute or measure. Simple grouping could be based on descriptive categorical values, such as by gender, age, nationality, job rank, and so on. More network-specific categories could be applied such as the number of an actor's ties. More intricate categories can be applied such as node-level social network measures. The user would bin the value by indicating a numeric range, resulting in bins and grouping the nodes according to their appropriate bin belonging.

Blockmaps are also effective in the examination of subgroups after the clustering has been performed. An analyst can transverse the hierarchical tree of groups and interactively transverse the hierarchy tree. The blockmap will show any corresponding groups and their associated data, via tile area-size, color, and directly via a mouse-over.

5 Demonstrative user session

This section presents an example walk-through of a user session, using *ORA-LITE (*ORA-LITE 2017; Carley et al. 2013) as the software platform host. The demonstration use-case is designed around exploring a 100-node, 20% density cellular network (Frantz and Carley 2005a, b). Figure 12 is the example dataset displayed as a node-link diagram. Within the *ORA-LITE node-link visualization feature, the blockmap tool is selected from the tools drop-down menu. The user is presented with a root-level the blockmap canvas, as show in Fig. 13 (alphabetic labels added for this discussion). This single root-level tile is comprised of all 100 nodes making up the dataset. The *ORA-LITE implementation presents eight entry fields, or display areas, which include: (A) menu, (B) Current Split dropdown field, (C) OnClick Split downdown field, (D) Tree Navigation button, (E) Spilt Bins value, (F) data canvas, (G) Color Legend, and (H) Selection Path. The initial data is segemented according to the closenessCentrality values of the 100 nodes in the dataset. This is controlled by the drop-down menu labeled Current Split (Fig. 13 label B).

The blockmap menu (display area A) provides the user with simple high-level navigation options and an instructive help feature. The Color Legend (display area G) provides a color legend and minimum and maximum range-setting for the color legend. The user adjusts the minimum and maximum values according to the desired granularity of the display colors desired, usually intending to show contrast among the tiles. Figure 14 shows the change in the display from Fig. 13 when the

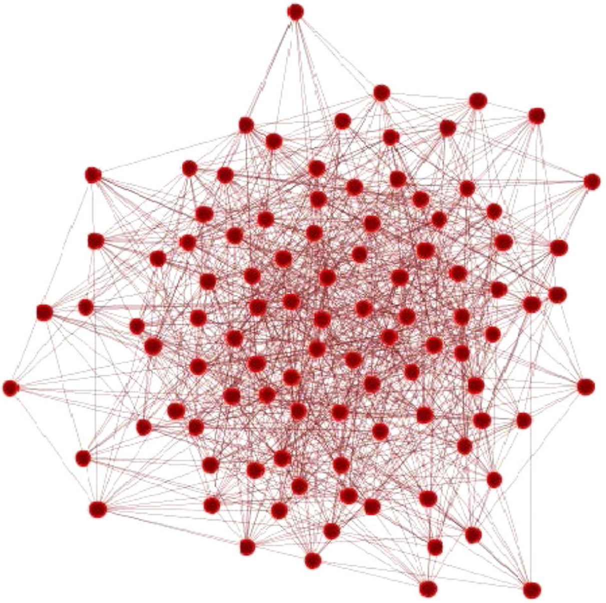
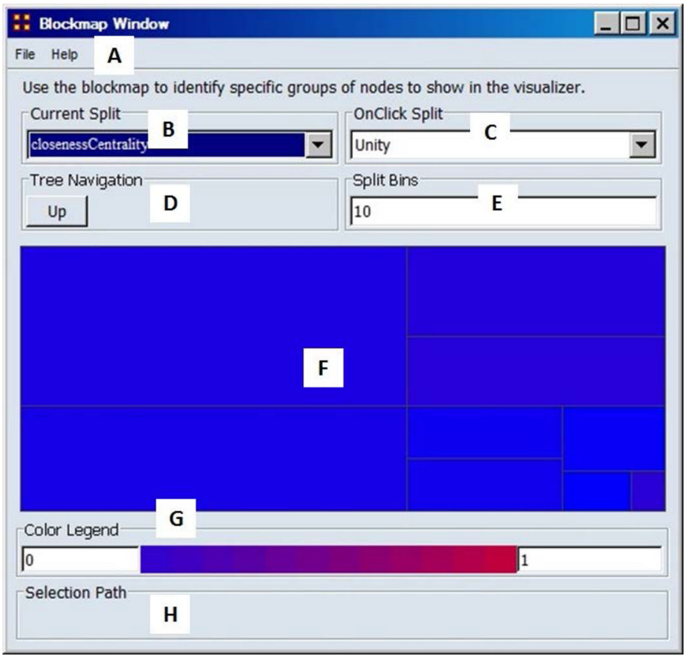


Fig. 12 Node-Link diagram of demonstration data consisting of a 100-node, 20% density undirected network



Use the blockmap to identify specific groups of nodes to show in the visualizer.

Current Split: closenessCentrality

OnClick Split: Unity

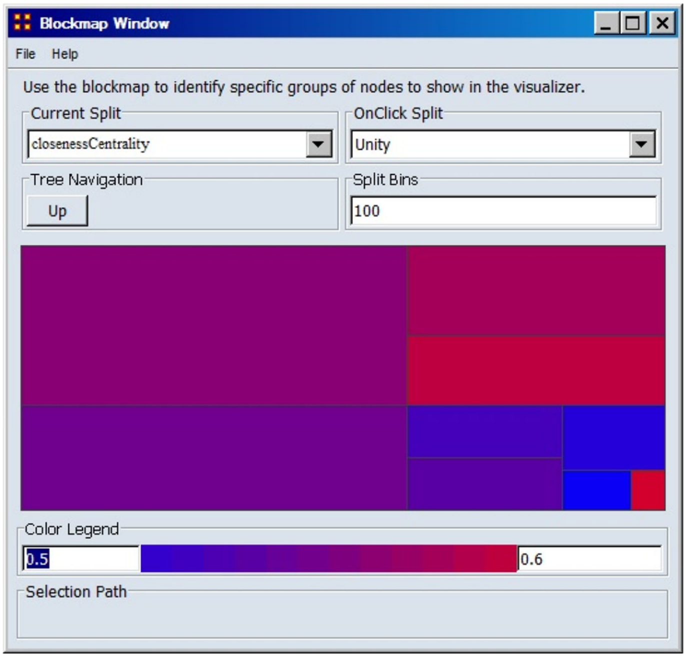
Tree Navigation

Split Bins: 10

Color Legend: 0 – 1

Selection Path

Fig. 13 Initial blockmap display of demonstration data—author-added labels overlying original image



Use the blockmap to identify specific groups of nodes to show in the visualizer.

Current Split: closenessCentrality

OnClick Split: Unity

Tree Navigation

Split Bins: 100

Color Legend: 0.5 – 0.6

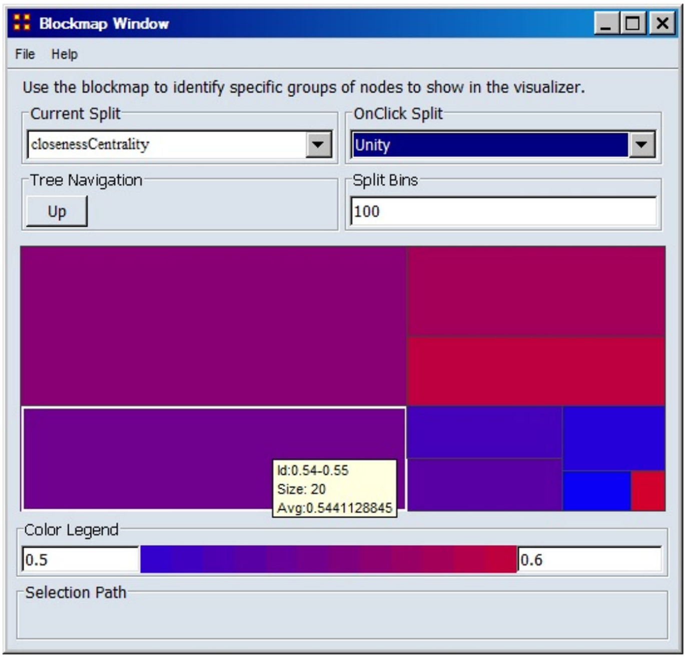
Selection Path

Fig. 14 Blockmap display after user changes legend values

Color Legend minimum and maximum values are changed to 0.5 and 0.6, respectively. The mouse-over display feature is activated on the lower-left tile and illustrated in Fig. 15. The mouse-over window presents selected attributes of the data comprising the collection of nodes represented by the tile. In this case making up this tile are 20 nodes, from the root-level population of 100 nodes, with closeness centrality between 0.54 and 0.55, with a mean value of 0.544.

Next for this demonstration, the user desires to explore the data presented by the bottom left box, i.e., 20 nodes with closeness centrality of 0.54–0.55. The user would like to see the drill down of the 20-nodes according to the betweennessCentrality measure, so she sets the OnClick Split dropdown field (Fig. 13 label C) to betweennessCentrality. The OnClick Split dropdown menu presents the same choices to the user as the Current Split dropdown menu (Fig. 13 label B).

Figure 16 shows the canvas after the user left-clicks the mouse, when held over the bottom left tile. Figure 17 shows the mouse-over information indicating that the single block represents the 20 nodes and the betweenness value ranges from 0.0 and 0.1 and that the mean value is 0.0063. The user would like to see more granularity in this data so she sets the Split Bins field (Fig. 13 label E) to 1000, indicating that the data being displayed should be split into bins that are based on betweenness centrality value ranges that are 1/1000 wide. Also, the user sets the Color Legend (Fig. 13 label G) values to 0.003 and 0.008. Figure 18 shows the result of these actions. Notice the 20-nodes are split into 4 blocks according to their betweenness centrality value with each block holding a range of 1/1000. This results in four



Use the blockmap to identify specific groups of nodes to show in the visualizer.

Current Split: closenessCentrality

OnClick Split: Unity

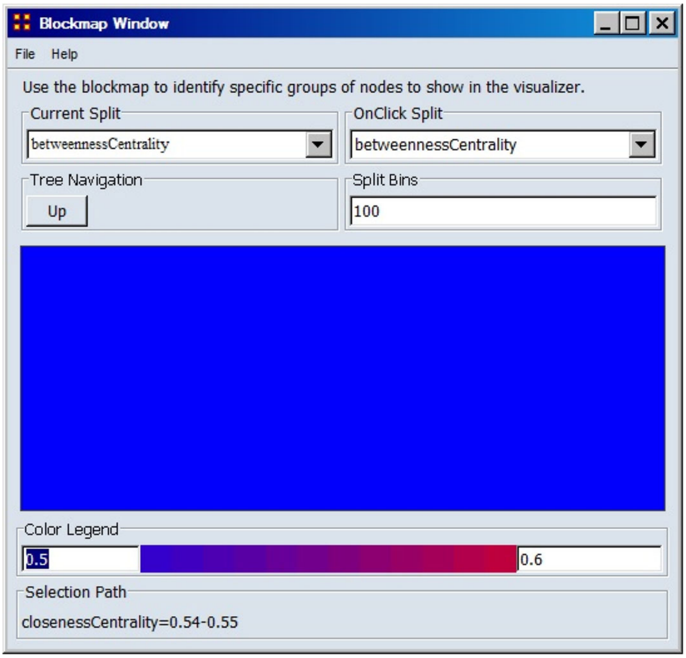
Split Bins: 100

Id:0.54-0.55
Size: 20
Avg:0.5441128845

Color Legend: 0.5 – 0.6

Selection Path

Fig. 15 Blockmap display after user moves mouse over a block



Use the blockmap to identify specific groups of nodes to show in the visualizer.

Current Split: betweennessCentrality

OnClick Split: betweennessCentrality

Split Bins: 100

Color Legend: 0.5 – 0.6

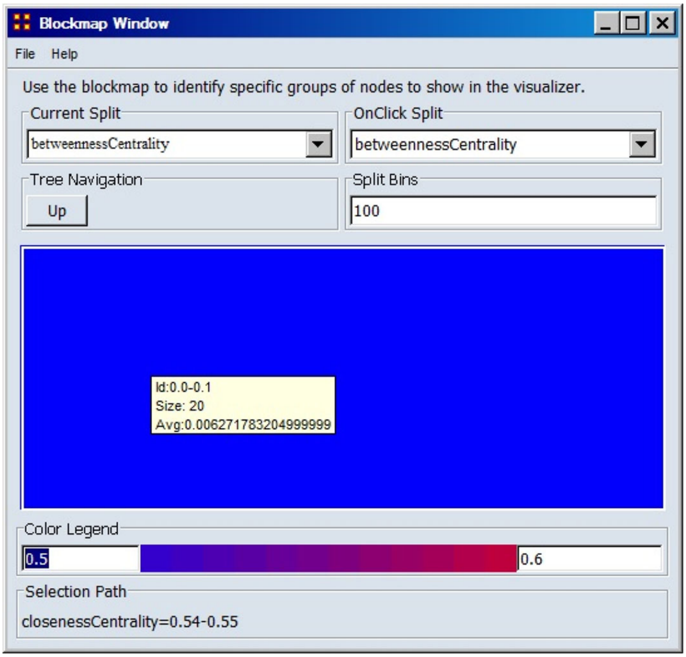
Selection Path: closenessCentrality=0.54-0.55

Fig. 16 Blockmap display after user sets OnClip Split to BetweennessCentralty and left-clicks the bottom-left block

blocks of nodes and as the mouse over shows the bottom left block represents six nodes with betweenness centrality of 0.006 to 0.007, with a block mean of 0.00653. Notice the Selection Path (Fig. 13 label H) display area on the bottom of the blockmap canvas. This provides the user with the previous (upper part of the tree) selections that we made, thus leading to the data presently being displayed.

Next in this demonstration, the user wishes to drill down into the six nodes represented by the bottom-left block. The user sets the OnClick Split dropdown menu to TotalDegreeCentrality (for example), sets the Split Bins to 10000, and the Color Legend to 0.17 and 0.21, then left-clicks the bottom-left block. Figure 19 shows the outcome of these field changes. Notice the Selection Path display displays the path taken, through the hierarchical tree, to reach this point. Figures 20 and 21 show the mouse-overs for the other two blocks to illustrate that the three blocks total to the prior six nodes (see Fig. 18). Finally for this demonstration, the user wishes to transverse up the tree and return to the parent block. Figure 22 shows the result of pressing the Tree Navigation UP button (Fig. 13 label D). Notice that the Selection Path display no longer shows the betweenness Centrality-0.0060–0.0070 string and now indicates only the Closeness Centrality = 0.654–0.55 string, reflecting that the tree has used this criteria to get to the data presently being displayed.

This demonstrative example intends to provide a simple primer for navigating around the *ORA_LITE implementation of the blockmap. The options and choices



Use the blockmap to identify specific groups of nodes to show in the visualizer.

Current Split: betweennessCentrality

OnClick Split: betweennessCentrality

Tree Navigation

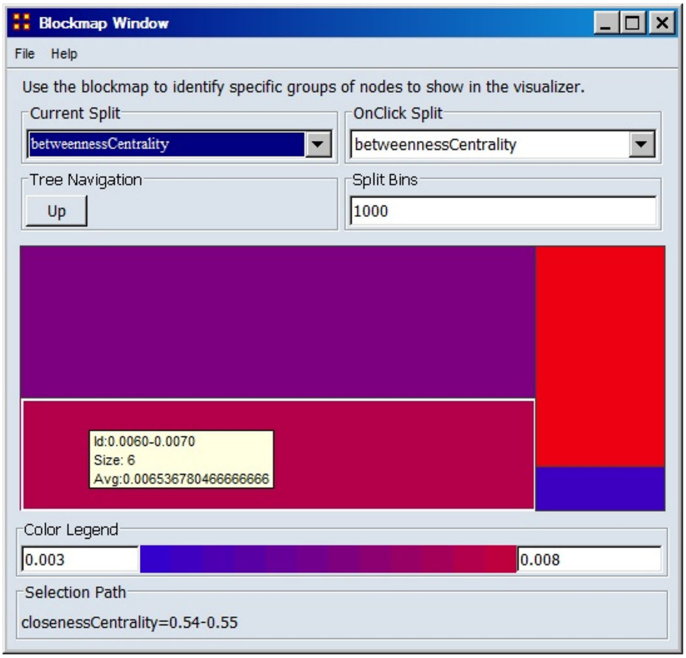
Split Bins: 100

Id:0.0-0.1
Size: 20
Avg:0.006271783204999999

Color Legend: 0.5 – 0.6

Selection Path: closenessCentrality=0.54-0.55

Fig. 17 Blockmap display after user moves mouse over the unity block



Use the blockmap to identify specific groups of nodes to show in the visualizer.

Current Split: betweennessCentrality

OnClick Split: betweennessCentrality

Tree Navigation

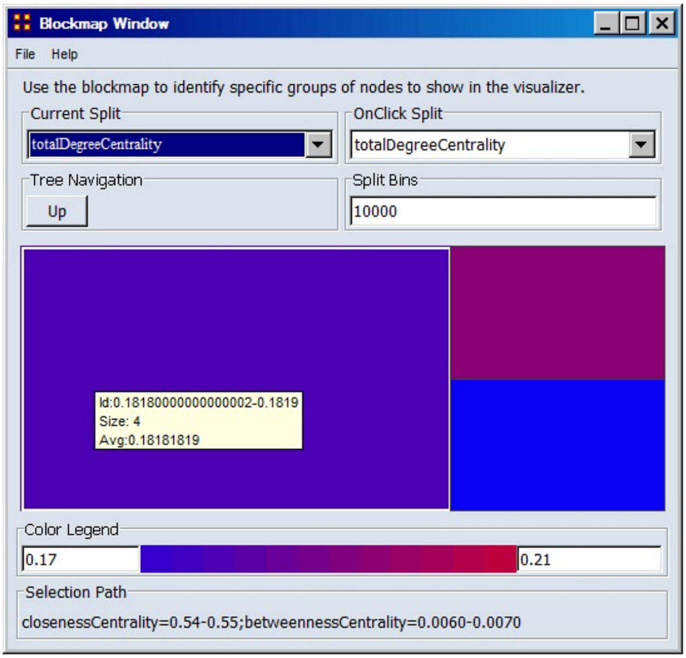
Split Bins: 1000

Id:0.0060-0.0070
Size: 6
Avg:0.006536780466666666

Color Legend: 0.003 – 0.008

Selection Path: closenessCentrality=0.54-0.55

Fig. 18 Blockmap display after user modifies the SplitBins and the legend values



Use the blockmap to identify specific groups of nodes to show in the visualizer.

Current Split: totalDegreeCentrality

OnClick Split: totalDegreeCentrality

Tree Navigation

Split Bins: 10000

Id:0.18180000000000002-0.1819
Size: 4
Avg:0.18181819

Color Legend: 0.17 – 0.21

Selection Path: closenessCentrality=0.54-0.55;betweennessCentrality=0.0060-0.0070

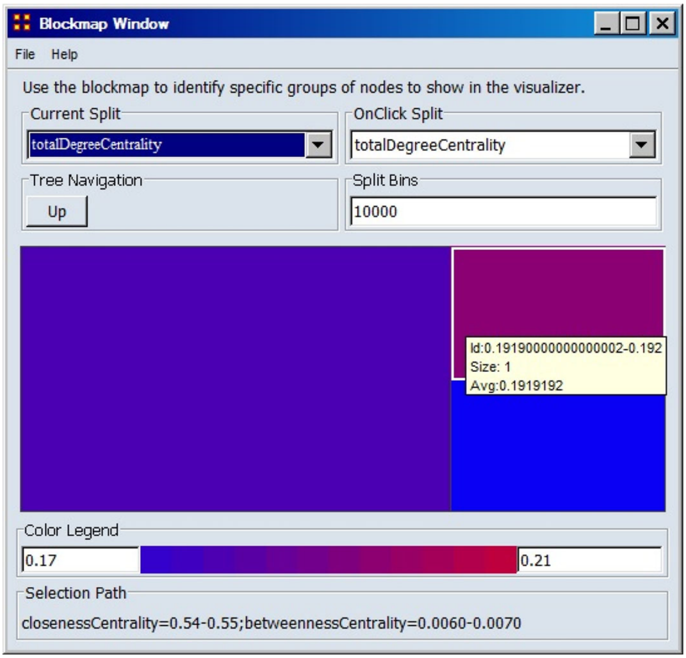
Fig. 19 Blockmap display after user modifies the OnClick Split, SplitBins and Legend values, then clicks mouse on lower-left block

that the user needs to understand and select from are rather few, so user-training time is limited; however, how the user applies these features is boundless.

6 Summary

Herein, the blockmap is introduced as a tool for exploring large datasets of relational network data. The technology is a combination of a treemap and heatmap, specifically applied to network data. The benefits of using a blockmap include being enabled to meaningfully explore big-data in a physically-fixed work area, in contrast to a node-link display. The information-loss accompanying the blockmap is limited to the specific dyads ties between the node; however, if this is a necessary requirement of the user, the relational links for each node, could be transformed into an attribute value and treated as any other attribute in the blockmap.

One extension to the blockmap display that would enhance the user experience would be to provide a tree display to accompany the blockmap display. This hierarchical tree display feature would draw the data tree that the user is essentially creating and traversing. Showing a display of the hierarchical tree (Sathiyanarayanan and Burlutskiy 2015a, b) would require more physical space on a display, but would significantly reduce the mental accounting that the user must maintain as the tree is being built and traversed. The Selection Path display in the present



Use the blockmap to identify specific groups of nodes to show in the visualizer.

Current Split: totalDegreeCentrality

OnClick Split: totalDegreeCentrality

Tree Navigation

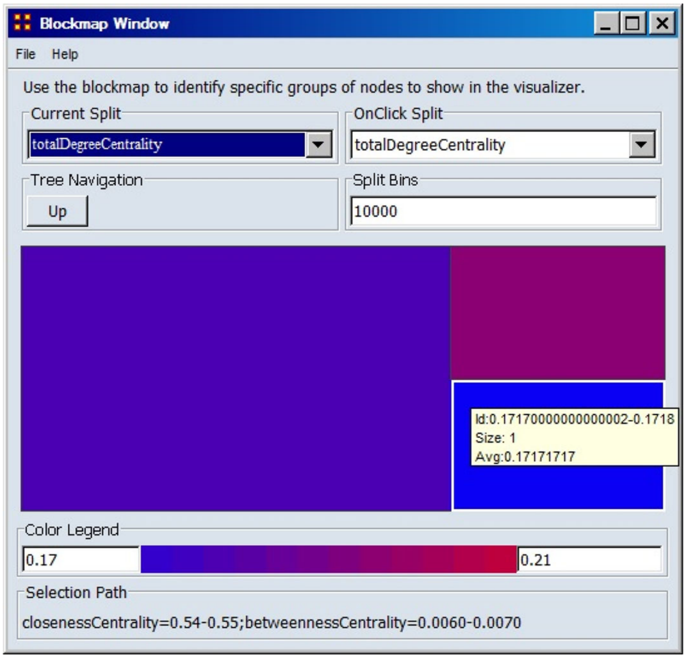
Split Bins: 10000

Id:0.19190000000000002-0.192
Size: 1
Avg:0.1919192

Color Legend: 0.17 – 0.21

Selection Path: closenessCentrality=0.54-0.55;betweennessCentrality=0.0060-0.0070

Fig. 20 Blockmap display after user places mouse over top right block



Use the blockmap to identify specific groups of nodes to show in the visualizer.

Current Split: totalDegreeCentrality

OnClick Split: totalDegreeCentrality

Tree Navigation

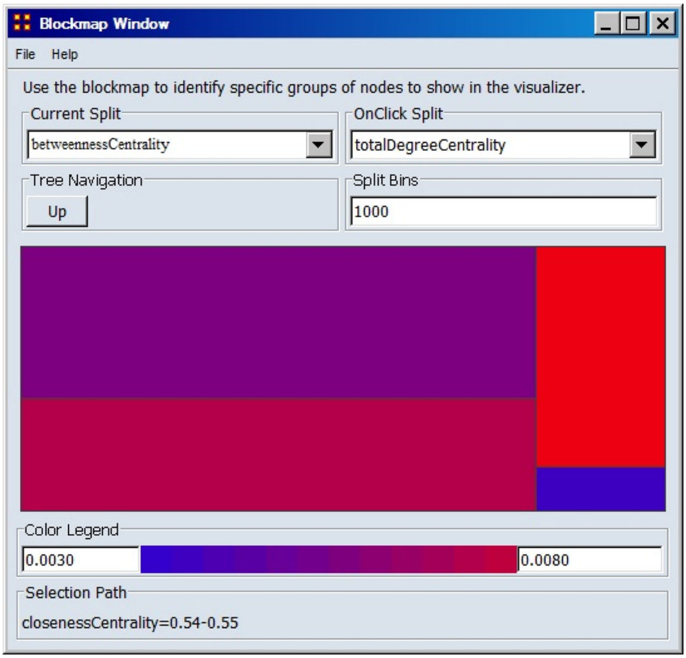
Split Bins: 10000

Id:0.17170000000000002-0.1718
Size: 1
Avg:0.17171717

Color Legend: 0.17 – 0.21

Selection Path: closenessCentrality=0.54-0.55;betweennessCentrality=0.0060-0.0070

Fig. 21 Blockmap display after user places mouse over bottom right block



Use the blockmap to identify specific groups of nodes to show in the visualizer.

Current Split: betweennessCentrality

OnClick Split: totalDegreeCentrality

Split Bins: 1000

Color Legend: 0.0030 – 0.0080

Selection Path: closenessCentrality=0.54-0.55

Fig. 22 Blockmap display after user clicks the Up button in TreeNavigation area

*ORA-LITE implementation accommodates this need, but a visualization display of the tree would be beneficial.

As the previous demonstration session indicated, blockmap technology is fully implemented in *ORA-LITE software; moreover, aspects of the technology are also implemented in numerous web-based utilities and commercial, statically programs. The website http://blockmap.net has additional informational links and a browser-based implementation which freely available. Software developers, especially those working with network data, are encouraged to develop blockmap technology for their platforms and programs.

References

*ORA-LITE (2017) Network analysis software. http://www.casos.cs.cmu.edu/projects/ora/software.php. Accessed 12 Jan 2017

4-traders.com (2017) Heatmap of S&P 500 index. http://www.4-traders.com/S-P-500-4985/heatmap-sectors/. Accessed 25 March 2017

Babaria K (2001) Introduction to Treemap. University of Maryland. http://www.cs.umd.edu/hcil/treemap3/TreemapIntroduction.pdf

Bederson BB, Shneiderman B (2003) The craft of information visualization: readings and reflections. Morgan Kaufmann

Blanch R, Lecolinet E (2006) Navigation techniques for zoomable treemaps. In: Adj Proc: Demos of ACM UIST'06, pp 49–50

Bruls M, Huizing K, van Wijk JJ (2000). Squarified Treemaps. In: Proceedings of VisSym '00 (May 2931, Amsterdam, The Netherlands). Springer, Wien, New York, pp 33–42

Carley KM, Pfeffer J, Reminga J, Storrick J, Columbus D (2013) ORA User's Guide 2013. Carnegie Mellon University Technical Report CMU-ISR-13-108

Cesarano A, Ferrucci F, Torre M (2016) A heuristic extending the Squarified treemapping algorithm. arXiv preprint arXiv:1609.00754

Chu H, Lin W, Wu J, Zhou X, Chen Y, Li H (2012) A new heat-map-based algorithm for human group activity recognition. In: Proceedings of the 20th ACM international conference on Multimedia. ACM, pp 1069–1072

Corominas-Murtra B, Goni J, Sole RV, Rodríguez-Caso C (2013) On the origins of hierarchy in complex networks. Proc Natl Acad Sci 110(33):13316–13321

de Nooy W, Mrvar A, Batagelj V (2005) Exploratory social network analysis with Pajek. Cambridge University Press, Cambridge

Fekete JD, Plaisant C (2002a) Interactive information visualization to the million. In: Symposium on information visualization (InfoVis' 02), Massachusetts, USA

Fekete J-D, Plaisant C (2002b) Interactive information visualization of a million items. In: Proceedings of IEEE symposium on information visualization 2002 (infoVis 2002), Boston, October

Frantz TL, Carley KM (2005) Treemaps as a tool for social network analysis. Carnegie Mellon University, School of Computer Science, CMU-ISRI-05-118

Frantz TL, Carley KM (2005) A formal characterization of cellular networks. Available at SSRN: https://ssrn.com/abstract=2726808

Freeman LC (2000a) Visualizing social groups. In: 1999 proceedings of the section on statistical graphics. American Statistical Association, pp 47–54

Freeman LC (2000b) Visualizing social networks. J Soc Struct 1(1):4

Friendly M (1994) Mosaic displays for multi-way contingency tables. J Am Stat Assoc 89(425):190–200

Frisch M, Heydekorn J, Dachselt R (2009) Investigating multi-touch and pen gestures for diagram editing on interactive surfaces. In: Proceedings of the ACM international conference on interactive tabletops and surfaces. ACM, pp 149–156

Holten D (2006) Hierarchical edge bundles: visualization of adjacency relations in hierarchical data. IEEE Trans Vis Comput Graph 12(5):741–748

Johnson B, Shneiderman B (1991) Tree-maps: a space-filling approach to the visualization of hierarchical information structures. In: Proceedings of the 2nd international IEEE visualization conference, pp 284–291, San Diego, CA

Keller R, Eckert CM, Clarkson PJ (2006) Matrices or node-link diagrams: which visual representation is better for visualising connectivity models? Inf Vis 5(1):62–76

Kolatch E, Weinstein B (2001) Cattrees: dynamic visualization of categorical data using treemaps. Project report

Kules B, Shneiderman B, Plaisant C (2003) Data exploration with paired hierarchical visualizations: initial designs of PairTrees. In: Proceedings of the 2003 annual national conference on digital government research. Digital Government Society of North America, pp 1–6

Lin W, Chu H, Wu J, Sheng B, Chen Z (2013) A heat-map-based algorithm for recognizing group activities in videos. IEEE Trans Circuits Syst Video Technol 23(11):1980–1992

Newman ME, Leicht EA (2007) Mixture models and exploratory analysis in networks. Proc Natl Acad Sci 104(23):9564–9569

Rouse M (2017) Treemap. http://searchenterprisedesktop.techtarget.com/definition/treemap. Accessed 4 April 2017

Saket B, Simonetto P, Kobourov S, Börner K (2014) Node, node-link, and node-link-group diagrams: an evaluation. IEEE Trans Vis Comput Graph 20(12):2231–2240

Sathiyanarayanan M, Burlutskiy N (2015) Design and evaluation of euler diagram and treemap for social network visualisation. In: 2015 7th international conference on communication systems and networks (COMSNETS). IEEE, pp 1–6

Sathiyanarayanan M, Burlutskiy N (2015b) Visualizing social networks using a treemap overlaid with a graph. Procedia Comput Sci 58:113–120

Shneiderman B, Plaisant C (1998) Treemaps for space-constrained visualization of hierarchies. http://www.academia.edu/download/30742877/shneiderman_treemap-history_1998-2009.pdf

Tukey JW (1977) Exploratory data analysis. Addison-Wesley, Reading

Viégas FB, Donath J (2004) Social network visualization: can we go beyond the graph? In: Workshop on social networks, CSCW'04, 6–10 November 2004. Chicago, IL

Ware C, Bobrow R (2005) Supporting visual queries on medium-sized node–link diagrams. Inf Vis 4(1):49–58

Wilkinson L, Friendly M (2009) The history of the cluster heat map. Am Stat 63(2):179–184

Terrill L. Frantz is an Assistant Professor of Management at Peking University HSBC Business School. His research specialty is in Organization Behavior during the Post-Merger Integration (PMI) stage of organizational mergers and acquisitions. He studies PMI using computational models of the organization based on social network constructs. Dr. Frantz has a life-long orientation in software technology, finance, and international business management, buttressed by both industry and academic experience.