gravicom - a web-based tool for community detection in networks

Andrea J. Kaplan

December 23, 2013

1 Background

1.1 Networks

1.2 Visualization

1.3 Layout Algorithms

2 User Interface

2.1 Design and Functionality

Before discussing gravicom's performance and use on example datasets, we give a brief overview of the components and functionality that make up the tool.

2.1.1 Description

The gravicom interface is comprised of five main parts,

1. Control panel
2. Data management
3. Connection table
4. Graph display
5. Tabset.

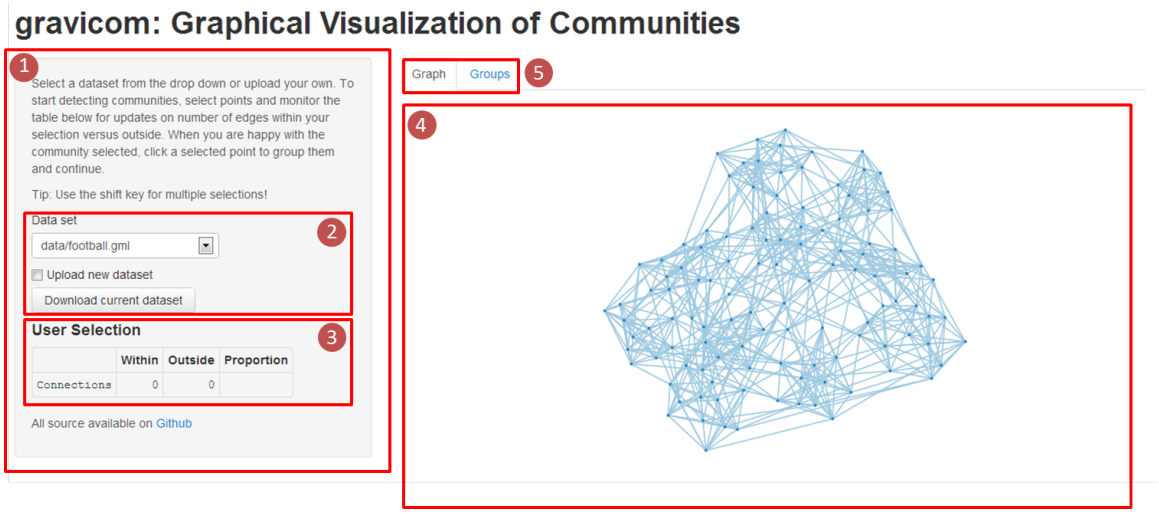


Figure 1: The components that make up gravicom, (1) Control panel, (2) Data management, (3) Connection table, (4) Graph display, and (5) Tabset.

Each part provides a means for the user to interact with gravicom, either through controls that allow user input to gravicom or through direct interaction with diagnostics and visualization of a graph. Their placement on the gravicom interface can be seen in figure **??**.

Control Panel The control panel serves as the starting point for a user's session in gravicom. It contains intructions for the user, as well as the means for a user to select a dataset and numerical summaries of the graph (such as a diagnostic connection table explained in ?? to follow). Additionally, the control panel contains a link to the source code for gravicom, should a user be interested in the inner workings of gravicom.

Data Management The data management component is made up of two main parts, data selection and data download. The data selection can be accomplished in two ways, the first being a drop down to select pre-loaded datasets to display. Currently there are two well-known network datasets provided in gravicom, a college football dataset [5] and a karate club dataset [**zachary1977information**]. From the dropdown the user can change the dataset to display in the graph. The second approach to data selection gives the user the ability to upload his own dataset. Upon clicking the "Upload new dataset" checkbox, a file selection control appears which gives the user the ability to upload his own graph data to explore with gravicom. This is shown in figure **??**.

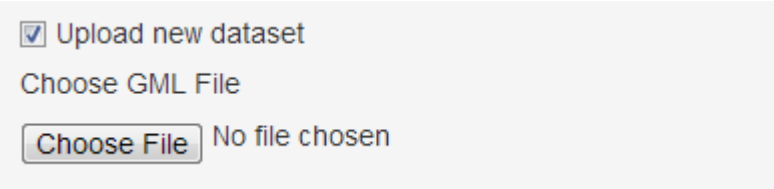


Figure 2: The data selection area upon clicking the "Upload new dataset" checkbox.

The work performed by a user in visualizing graphs and community structure can also be downloaded as a dataset from gravicom with current communities stored. This feature can be used as a

save point in processing a graph or as a means to export changes made in gravicom to another tool.

Connection Table The connection table is a quantitative diagnostic tool for the user in assessing the strength of a community structure in a graph. The idea behind the connection table is that a community of nodes will have proportionally more edge connections within the node cluster compared to edge connections to nodes outside the community. The table displays the number of edge connections within a user's selection of nodes in the graph and the number of connections from nodes in a user's selection to nodes not in the selection. The comparison of these two numbers can give the user a rough idea of if the plausibiity or extend to which the node selection constitutes a community. To aid in the comparison, there is also a proportion column that displays the ratio of number of connections within a node selection to the number of connections outside the selection.

Graph Display The graph display shows an interactive graphical representation of the selected (or uploaded) graph data. Upon load, the graph displays all nodes and edges in the dataset using a force-directed layout algorithm. The user has several ways to interact with the graph: drag, select, and group. A user can drag a node at any time. Upon dragging, the force-directed layout is rerun, giving an altered view of the graph. Figure **??** shows a graph in the process of being dragged.

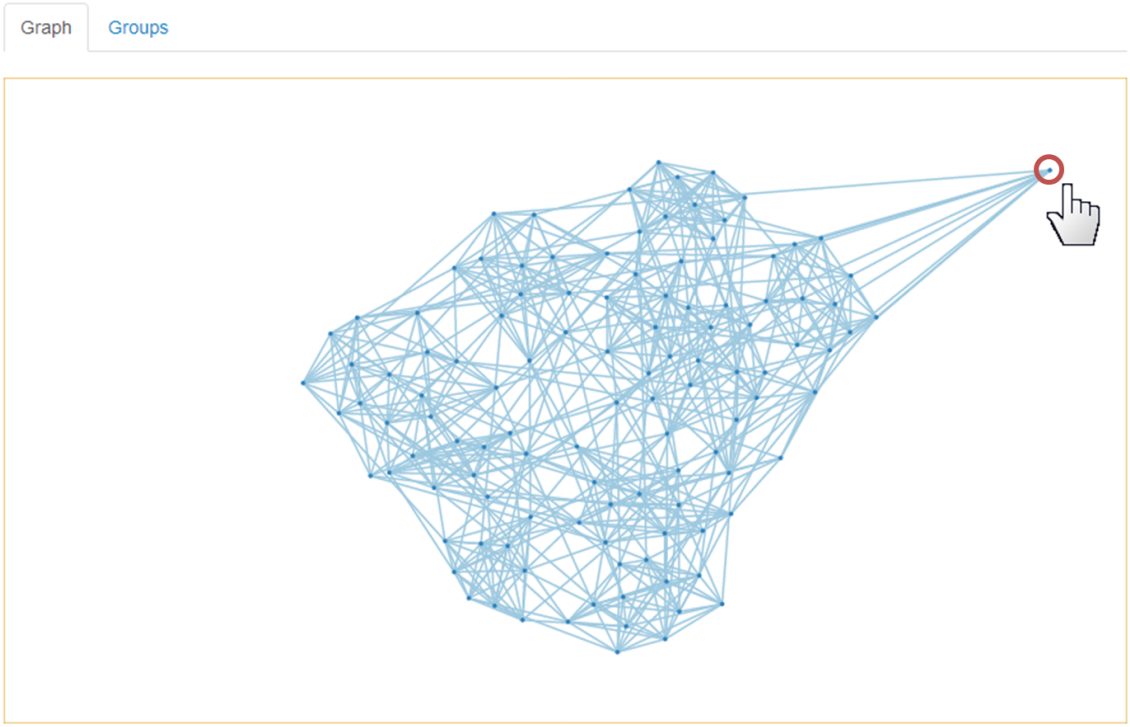
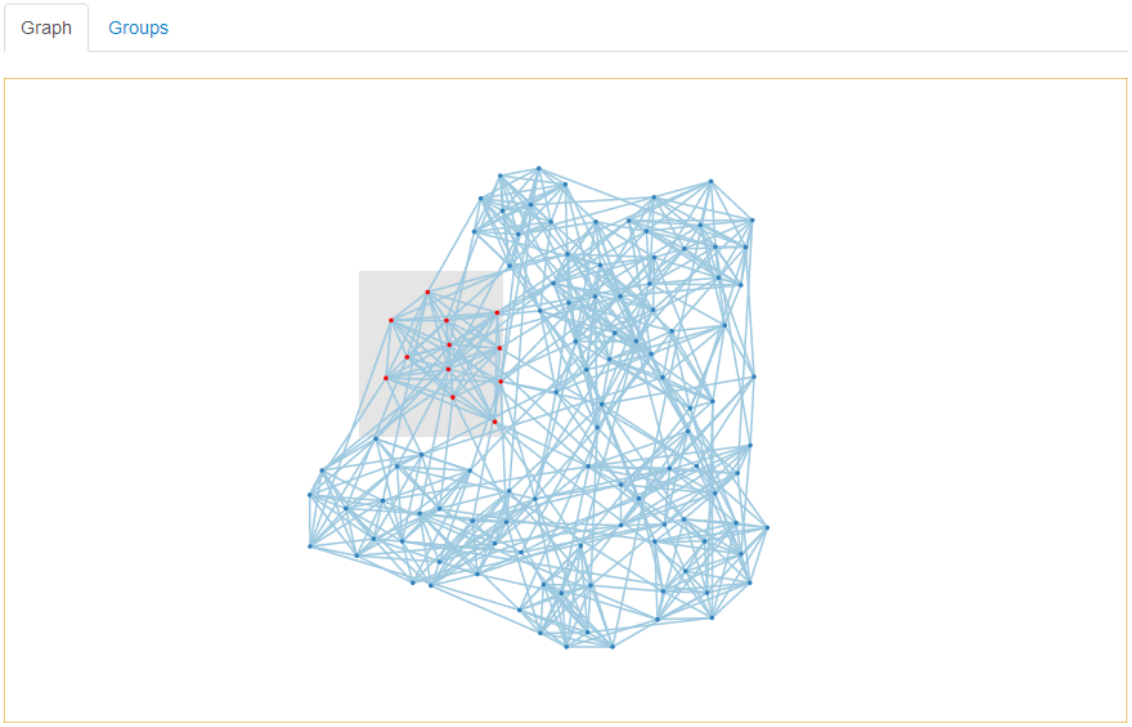


Figure 3: A graph in the process of being dragged. The node being dragged is marked by a red circle.

Selection and grouping of nodes are actions intended to work todether. In order to group nodes, a user first determines a node cluster or potential community based on a visual appraisal of the graph. To select nodes the user clicks and drags a selection box around nodes. See figure **??** for the results of selection in the interface. The shift key can also be used for multiple selections. Upon

selection, the connection table is updated and the user can evaluate the selection as a community and alter the selection if need be (the shift key selection is useful in this step).



User Selection

	Within	Outside	Proportion
Connections	46	43	1.0698

Figure 4: A graph in the process of nodes being selected. Upon selection of nodes, the connection table updates to display within and outside edges.

Selected nodes can be grouped together into one consolidated "super-node" or grouped-node. Once grouped, a new node is created that comprises all grouped nodes and grouped within edges in size and charge. The force-directed layout is again run, showing the new graph with previous nodes grouped. This is illustrated in figure **??**. This process of node grouping can be repeated until all nodes have been grouped or until the user is satisfied that all communities have been selected. Additionally, grouped nodes can be ungrouped by clicking on a grouped node.

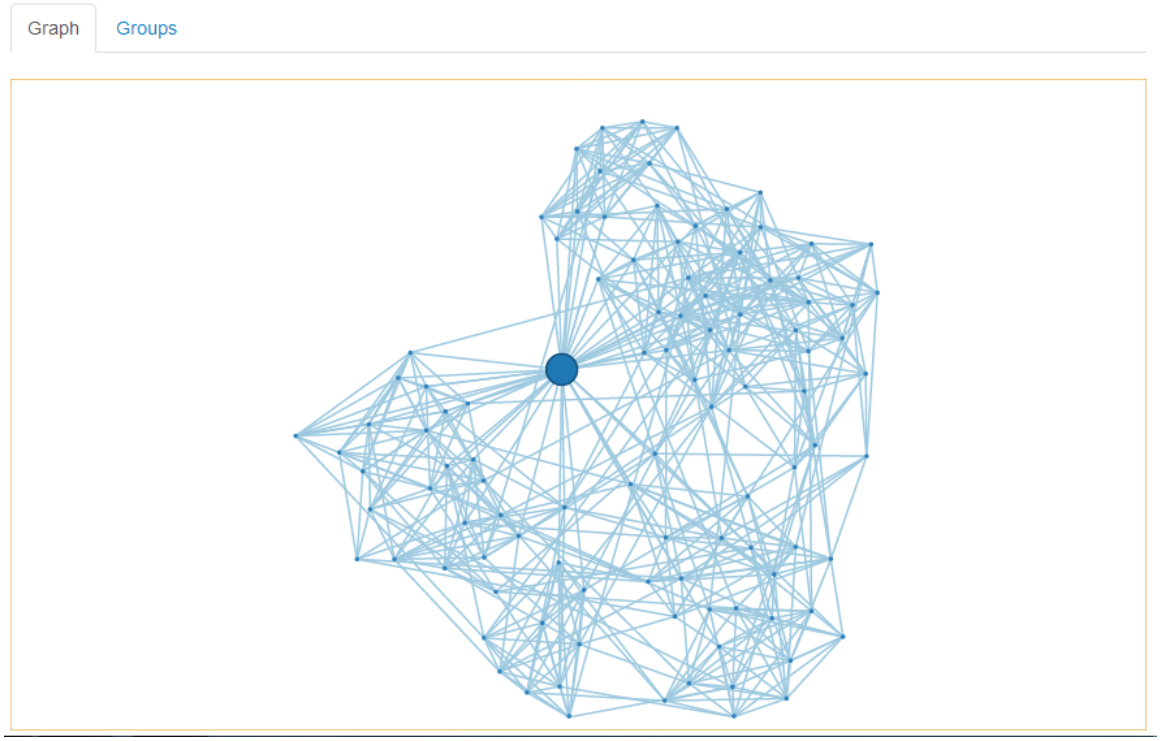


Figure 5: A graph after nodes have been grouped and the force-directed algorithm has been re-run.

Tabset The tabset allows the user to switch between two tabs on the screen. The first (and default tab) shows the graph display. The second tab shows the groups that a user has created in the graph. Within this tab, each node group or community summarizes the number of nodes in the group and also provides the ability to drop down and view the node IDs for that group. If the data are equipped with node labels, these will be displayed. If there are no node labels provided, node IDs will be shown as node numbers. For an example, see figure **??**.

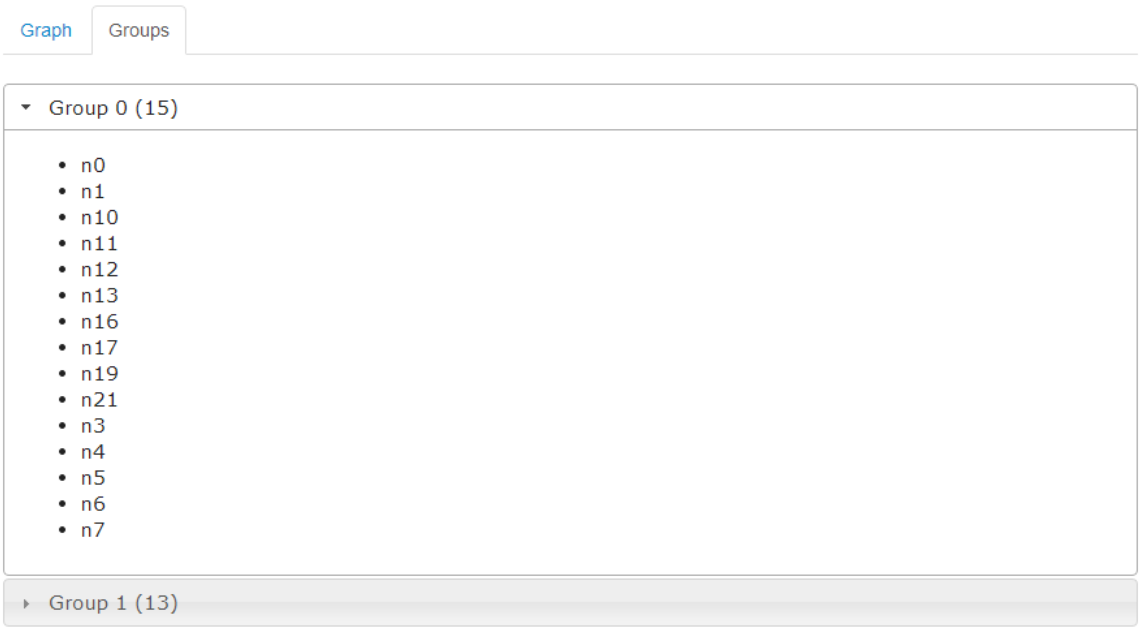


Figure 6: The groups tabset displaying which nodes have been grouped in Group 0, for example. The groups tabset also shows that Group 0 has 15 nodes, while Group 1 has 13 nodes.

3 Examples

To demonstrate the use of gravicom, we present three real-world network datasets and explore their community structure.

3.1 College Football

The first dataset is a representation of U.S. College Football Division 1 games from the 2000 season [5]. This is a default dataset available in gravicom. In this network, nodes represent teams and an edge represents a regular-season game played between the two connected teams. The distances between the nodes are based on the number of games played between the teams. The network as it appears upon load in gravicom is presented in figure **??**.

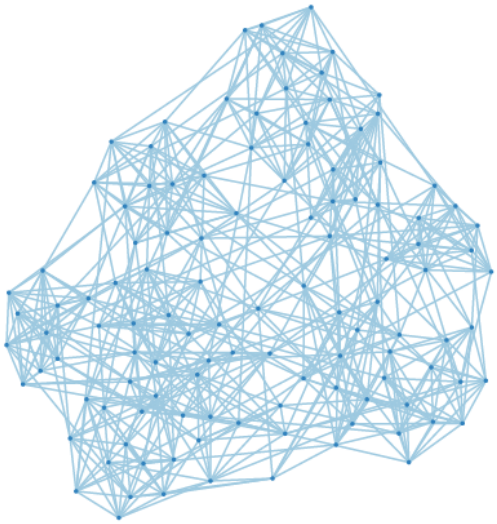
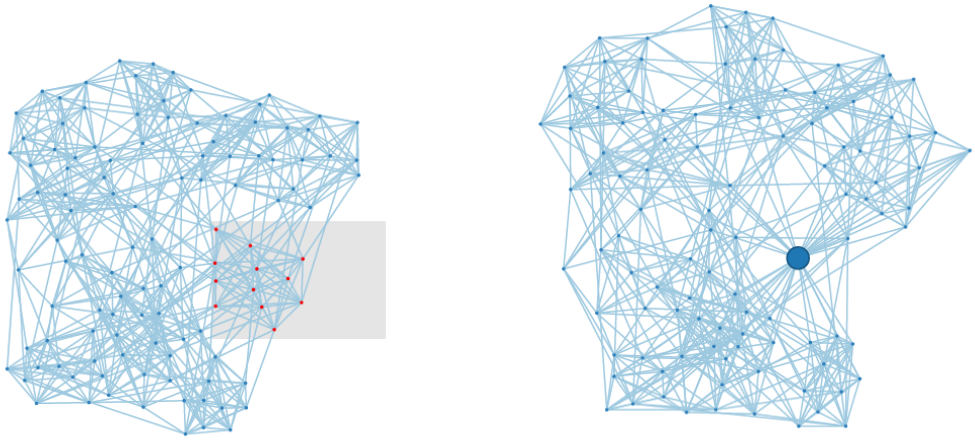


Figure 7: College Football network represented in gravicom. Communities are visually evident and will become more identifiable as other are grouped.

Colleges within the same football conference will play members of their conference more frequently than teams outside of their conference, making this an interesting dataset for attempting to visually detect community structures. This is also an ideal illustrative example due to the relatively small number of nodes and edges present, making a graphical representation particularly feasible. One challenge with this dataset, however is the existence of independent teams, like Notre Dame, which do not belong to any conference and so may complicate efforts of community detection. Another complication is that small conference schools typically play large conference schools at the beginning of a season in order to help fund their athletic programs, which can potentially cause more edges than perhaps expected between distinct communities, particularly between small conferences and large conferences.

Upon viewing the network in gravicom, some likely communities visually emerge, and by selecting nodes to examine within and outside edges, we can classify colleges into conferences, as seen in figure **??**.



User Selection

	Within	Outside	Proportion
Connections	44	36	1.2222

Figure 8: Selecting a potential community in gravicom and assessing the number of within verses outside selection edges. After the first community is detected, more communities become apparent in the network.

Once the user is satisfied that he has selected a viable community, he clicks to group those nodes and the graph will update to reflect this. In particular, the grouped nodes for the suspected community will be collapsed into one super-node in the updated graph ??. The resulting graph update allows new community structures to potentially become more easily apparent as seen in figure **??**.

Additionally, the user can check which nodes were grouped in each community ??. In this example (figure **??**), we can see the first community detected corresponds to the Big 10.

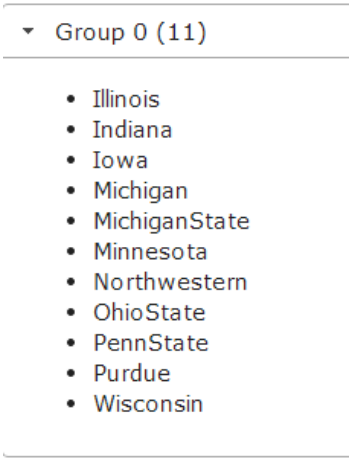


Figure 9: A drill-down of the first grouped community which corresponds to the Big 10 Conference.

After the first community has been selected, another potential community at the top of the graph has been revealed (explain). The user can once again select and group this node cluster into a community and subsequently examine the nodes in the resulting group as seen in figure **??**. The second conference grouped corresponds exactly to the SEC Conference, another large college football conference. Here, the first two communities to become evident correspond to large Division 1 conferences that play the majority of their games within conference, matching our earlier assertion that small conferences should, as expected, be more difficult to detect.

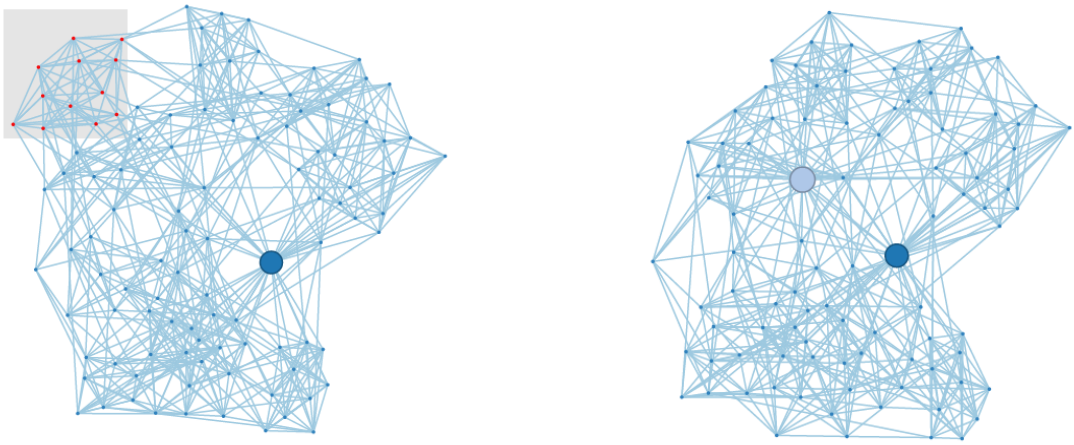


Figure 10: A second community is grouped which corresponds to the SEC Conference.

This process of grouping nodes into suspected communities can be repeated until the user is satisfied with the communities selected. The entirety of this process is seen in figure **??** and the resulting communities in table **??** (explain).

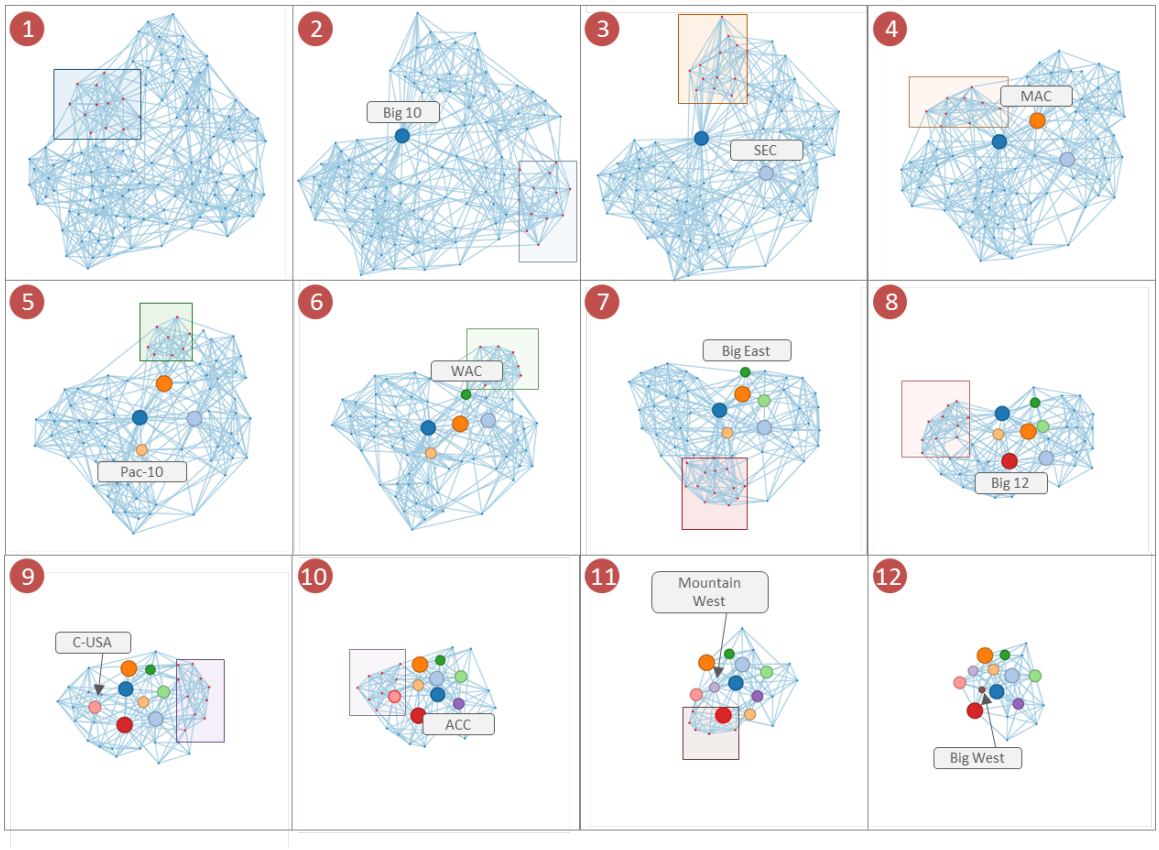


Figure 11: The full process of selecting and grouping communities in gravicom using the football dataset.

Conference	Teams Identified	Accuracy
ACC	Duke, Wake Forest, Virginia, Florida State, Clemson, North Carolina, Maryland, Georgia Tech, North Carolina State	100%
Big East	Boston College, Miami Florida, Virginia Tech, Syracuse, Temple, West Virginia, *Connecticut*, Pittsburgh, Rutgers	88.9%
Big 10	Ohio State, Penn State, Michigan, Michigan State, Purdue, Minnesota, Northwestern, Illinois, Iowa, Wisconsin, Indiana	100%
Big 12	Kansas State, Iowa State, Kansas, Texas A& M, Texas Tech, Baylor, Missouri, Texas, Oklahoma State, Colorado, Oklahoma, Nebraska	100%
C-USA	Cincinnati, Louisville, Houston, Tulane, Southern Mississippi, Army, Memphis, East Carolina, Alabama Birmingham	100%
Independent	Notre Dame, Navy	100%
MAC	*Central Florida*, Western Michigan, Miami Ohio, Ohio, Bowling Green State, Marshall, Ball State, Akron, Buffalo, Northern Illinois, Eastern Michigan, Toledo, Central Michigan, Kent	92.9%
Mountain West	Brigham Young, San Diego State, *Boise State*, Wyoming, New Mexico, Nevada Las Vegas, Utah, *North Texas*, *Utah State*, *New Mexico State*, Colorado State, *Arkansas State*, *Idaho*, Air Force	57.1%
Pac-10	Arizona, Oregon State, Washington, Washington State, Arizona State, UC LA, Stanford, Southern California, Oregon, California	100%
SEC	Vanderbilt, Florida, Louisiana State, South Carolina, Mississippi, Arkansas, Auburn, Kentucky, Georgia, Mississippi State, Alabama, Tennessee	100%
Big West	Middle Tennessee State, Louisiana Lafayette, Louisiana Monroe, *Louisiana Tech*	75%
WAC	Nevada, Fresno State, *Texas Christian*, Tulsa, Hawaii, Rice, Southern Methodist, San Jose State, Texas El Paso	88.9%

Table 1: Resulting communities detected using gravicom and the corresponding conference. Teams that have been incorrectly classified are italicized.

We detected visually 11 conferences in the dataset. There were 11 conferences and 5 independent schools in the dataset. Through manual specification of conferences, we were able to correctly classify 91.3% of the football teams into their conferences. (explain meaning of size/distance etc.)

3.2 Political Books Sold

For illustration, we next consider a second example dataset consisting of a network of political books purchased close to the 2004 United States presidential election and sold on Amazon.com [1]. Each node represents a book and each edge represents frequent copurchasing of two books by the same buyers. The books are classified as being conservative, liberal, or neutral by the author of the dataset and we can see a clear partition in figure **??** between two main groups (i.e. roughly conservative and liberal) with a smaller group between, when looking at the network in gravicom. This dataset is interesting because it highlights a great divide in the United States population that can be ascribed to the two party political system.

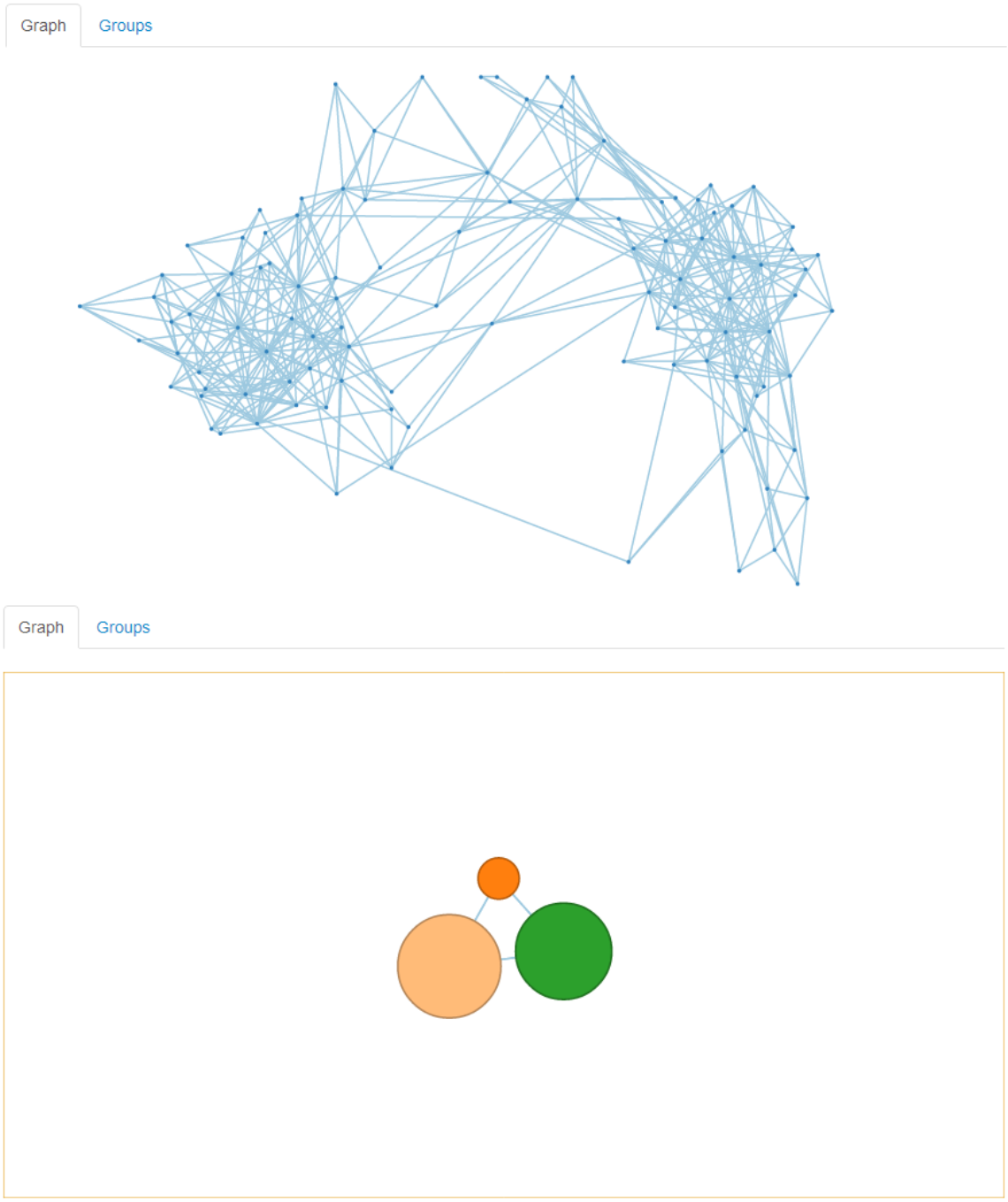


Figure 12: Political books dataset as seen in gravicom prior and post community detection.

Classification	Books Identified	Accuracy
Conservative	A National Party No More, Dereliction of Duty, Ten Minutes from Normal, Bush Country, Rumsfeld's War, Legacy, Hating America, Hillary's Scheme, *Meant To Be*, Tales from the Left Coast, Breakdown, Losing Bin Laden, The French Betrayal of America, Spin Sisters, The Right Man, Useful Idiots, Shut Up and Sing, Who's Looking Out for You?, Those Who Trespass, Bias, The O'Reilly Factor, Let Freedom Ring, Deliver Us from Evil, Give Me a Break, Betrayal, The Real America, The Faith of George W Bush, The Death of Right and Wrong, *Power Plays*, Arrogance, *The Perfect Wife*, The Bushes, Things Worth Fighting For, Off with Their Heads, Persecution, Why Courage Matters, Hollywood Interrupted, The Enemy Within, We Will Prevail, Endgame, The Official Handbook Vast Right Wing Conspiracy, The Third Terrorist, Slander, The Savage Nation, Fighting Back	93.3%
Liberal	Downsize This!, The Culture of Fear, House of Bush, House of Saud, The Best Democracy Money Can Buy, Rogue Nation, Stupid White Men, Rush Limbaugh Is a Big Fat Idiot, The Great Unraveling, Against All Enemies, American Dynasty, The Price of Loyalty, The Sorrows of Empire, Worse Than Watergate, *Plan of Attack*, Big Lies, The Lies of George W. Bush, Bushwomen, The Bubble of American Supremacy, Living History, The Politics of Truth, Fanatics and Fools, Bushwhacked, Disarming Iraq, Lies and the Lying Liars Who Tell Them, MoveOn's 50 Ways to Love Your Country, The Buying of the President 2004, Perfectly Legal, *Bush at War*, The New Pearl Harbor, Freethinkers, Had Enough?, It's Still the Economy, Stupid!, We're Right They're Wrong, What Liberal Media?, The Clinton Wars, Weapons of Mass Deception, Dude, Where's My Country?, Thieves in High Places, Shrub, Buck Up Suck Up, Hegemony or Survival, The Exception to the Rulers	95.2%
Neutral	1000 Years for Revenge, *Bush vs. the Beltway*, *Charlie Wilson's War*, *Dangerous Dimplomacy*, Sleeping With the Devil, *The Man Who Warned America*, Why America Slept, Ghost Wars, Surprise, Security, the American Experience, *Allies*, *The Choice*, All the Shah's Men, *Soft Power*, Colossus, The Future of Freedom, *Rise of the Vulcans*, *America Unbound*, Empire	50%

Table 2: Resulting communities detected using gravicom and the corresponding type of book. Books that have been incorrectly classified are italicized.

We detected visually 3 types of books in the dataset (how?). Through subsequent manual verification of the classification of books, we found that we were able to correctly classify 86.67% of the books into the categories created by the author of the dataset. See table **??** for the final groups of books found using gravicom.

3.3 Technical Aspects

gravicom utilizes three main pieces of software to establish interactive user control of a random graph as sketched out in figure **??**, which are Shiny, D3, and igraph. These are used, respectively, for server/client interaction management, user interface and graph layout, and data formatting, respectively. In the following subsections, we describe the purposes of these three components in

more detail.

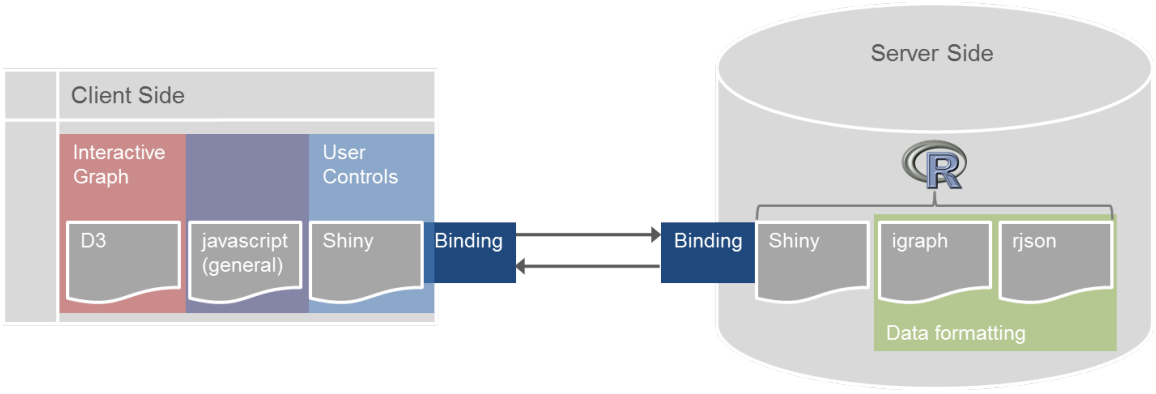


Figure 13: Relationship between client and server, specifically focusing on how data travels between the two.

There are very minimal software requirements for a user of gravicom. The client simply needs to have a JavaScript enabled internet browser with HTML5 compatibility, something which almost any modern browser fulfills (an exception is IE8 and below).

The server side requirements are more extensive, but this does not affect the user of gravicom, only those wanting to host their own instance of the application. To host gravicom, a Linux server is required, with the following installed:

- Node.js (0.8.16 or later)
- `R` (2.15 or later)
- Shiny `R` package, installed into the machine-wide site library.
- Shiny Server

3.3.1 Shiny

Shiny [6] is an `R` package created by RStudio that enables `R` users to create an interactive web application that utilizes `R` as the background engine. Through default methods to build user interface elements in HTML and a handle to the server side code, Shiny is a simple way to turn `R` code into a website.

gravicom uses the Shiny functionality to create user controls, pass correctly formatted data to the client, and as a means to display summary information regarding the user's interactions with a graph at any point in time. In this context, Shiny serves as the translator between the formatted data and what the user sees and interacts with on his screen.

3.3.2 D3

D3 [2] stands for "Data Driven Documents" and is a JavaScript library developed and maintained by Mike Bostock with the purpose of visualizing and interacting with data in a web-based interface. It is freely available from `http://www.d3js.org`. The library facilitates manipulation of HTML elements, SVG (scalable vector graphics), and CSS (cascading style sheets) with the end goal of rendering animations and providing user interactions that are tied to the underlying data. The key idea behind the library is that Document Object Model elements are completely determined by the

data. The Document Object Model (DOM) is a convention for representing and interacting with objects in HTML, XHTML and XML. So, rather than adding elements to a web page to be viewed by users, D3 allows users to see and interact with graphical representations of their data in a web framework.

gravicom uses D3 to handle all graphical displays and user interactions with the graph. The data is passed to the client and able to be used through Shiny's input bindings. It is crucial that the data has been formatted correctly at this point for the JavaScript to properly function. For this reason, we limit the file types being passed into the tool to a robust graph-specific type.

At this point in the page lifecycle, the graph nodes are tied to circles and the edges are tied to paths on the page. User manipulations such as selecting, dragging, and grouping are handled by D3 and data is passed back to the server via Shiny's output bindings to allow for communication between user and the `R` engine underneath. This is illustrated in figure **??**. What this means is that all visualization and user interaction with the graph are accomplished using JavaScript, more specifically the library D3. Shiny and `R` serve as the framework on which the data sits, but when the user touches the data they are doing so through the JavaScript elements.

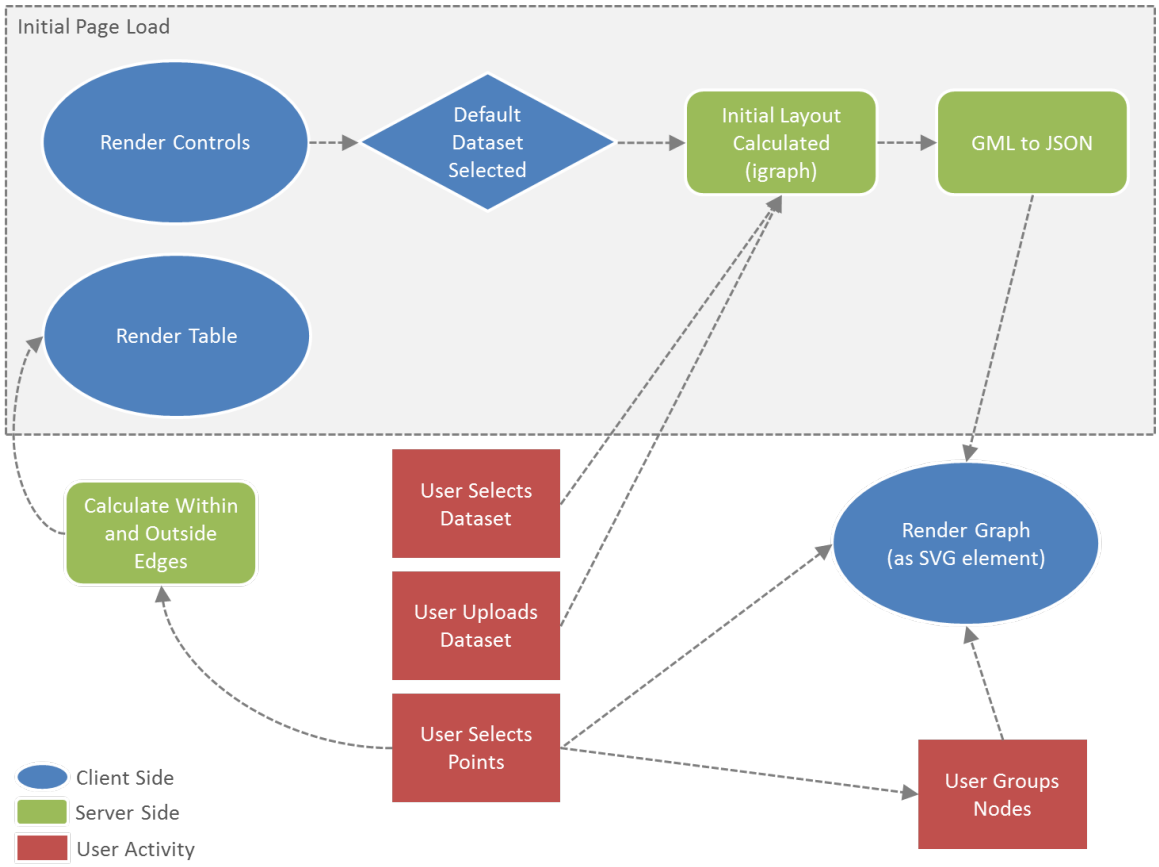


Figure 14: Page lifecycle beginning from on load. User actions are highlighted in red, server actions in green, and actions completed on the client side are highlighted in blue.

3.3.3 igraph

igraph [4] is a software package used for creating and manipulating undirected and directed graphs. It is a cross-language package available for C, R, python, and Ruby. igraph also supports multiple graph file formats and visualization of graph structures.

gravicom utilizes two parts of igraph, first is the conversion from a gml file to an XML file. The gml file format, short for Graph Modelling Language, is a hierarchical ASCII-based file format for describing graphs. Below is an example gml file of an undirected graph consisting of two nodes linked by a single edge. The important points to note are that node identifiers (id) have to be numeric. An edge consists only of source and target ids of the nodes it connects, while nodes can have other attributes, e.g. `value` in the example. For a directed graph, the parameter `directed` has to be set to 1, which will result in the edge information on target and source being evaluated accordingly.

```
## graph
## [
##   directed 0
##   node
##   [
##     id 0
##     label "Node 1"
##     value 100
##   ]
##   node
##   [
##     id 1
##     label "Node 2"
##     value 200
##   ]
##   edge
##   [
##     source 1
##     target 0
##   ]
## ]
```

For the conversion from an XML file to a JSON file we make user of the R package `rjson` [3]. JSON is the native data format used in D3, which makes working with data in the D3 library incredibly straightforward. Here is our example in the finalized JSON format:

```
## {
##   "nodes":
##   [{"id":"n0","v_id":"0","v_label":"Node 1","v_value":"100"},
##    {"id":"n1","v_id":"1","v_label":"Node 2","v_value":"200"}],
##  "edges":
##   [{"source":0, "target":1}]
## }
```

Our example data will yield the graph in figure **??**.

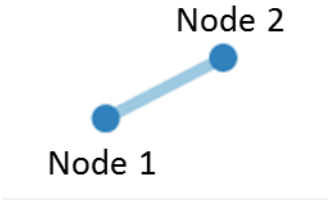


Figure 15: Graph created from sample gml file.

The second use of igraph within gravicom is to compute initial x and y coordinates for the nodes of the graph using a force-driven layout. This provides the initialization for the force-layout algorithm in D3. This reduces the computational load on the clients' side and helps minimize unnecessary movement by the nodes. This is critical as the extra movement at the loading of the pages creates an unnecessarily chaotic start to the user's experience.

4 Further Work

References

[1] *Books about US Politics*. http://networkdata.ics.uci.edu/data.php?d=polbooks.

[2] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. "D3: Data-Driven Documents". In: *IEEE Trans. Visualization & Comp. Graphics (Proc. InfoVis)* (2011). URL: http://vis.stanford.edu/files/2011-D3-InfoVis.pdf.

[3] Alex Couture-Beil. *rjson: JSON for R*. R package version 0.2.12. 2013. URL: http://CRAN.R-project.org/package=rjson.

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

[5] M. Girvan and M. E. J. Newman. "Community structure in social and biological networks". In: *Proceedings of the National Academy of Sciences* 99.12 (2002), pp. 7821–7826. DOI: 10.1073/pnas.122653799. eprint: http://www.pnas.org/content/99/12/7821.full.pdf+html. URL: http://www.pnas.org/content/99/12/7821.abstract.

[6] RStudio Inc. *shiny: Web Application Framework for R*. R package version 0.4.0. 2013. URL: http://CRAN.R-project.org/package=shiny.