

Are We in Agreement? Benchmarking and Reliability Issues between Social Network Analytic Programs

Philip J. Murphy
*Middlebury Institute of
International Studies
Monterey, CA, USA*

YuFei Wang
*Middlebury Institute of
International Studies
Monterey, CA, USA*

Karen T. Cuenco
*University of Pittsburgh
Pittsburgh, PA, USA*

Abstract

Reliability and validity are key concerns for any researcher. We investigate these concerns as they apply to social network analysis programs. Six well-used and trusted programs were compared on four common centrality measures (degree, betweenness, closeness, and eigenvector) under a variety of network topographies. We identify notable inconsistencies between programs that may not be apparent to the average user of these programs. Specifically, each program may have implemented a variant of a given measure without informing the user of its characteristics. This presents an unnecessary obfuscation for analysts seeking measures that are best suited to the idiosyncrasies of their data, and for those comparing results between programs.

Under such a paradigm, the terms in use within the social network analysis community become less precise over time and diverge from the original strength of network analysis: clarity.

Acknowledgements

The authors would like to thank Elma Paulauskaite and Maizy Cuenco for their help in putting together this research.

This work was funded in part through a grant from the National Institutes of Health: Grant number NIDCR 1R03DE020127.

Authors

Philip J. Murphy, Middlebury Institute of International Studies, Monterey, CA, USA.

YuFei Wang, Middlebury Institute of International Studies, Monterey, CA, USA.

Karen T. Cuenco, Department of Human Genetics, University of Pittsburgh, PA, USA.

Correspondence concerning this work should be addressed to Philip J. Murphy, Middlebury Institute of International Studies, 460 Pierce St., Monterey CA 93940 USA, pjmurphy@miis.edu; phone: 831-647-4600; fax: 831-647-6693

Introduction

An important part of the appeal of social network analysis originates from the incorporation of mathematical descriptions for social relations. This has made it possible to provide clear and unambiguous definitions for concepts relating to relational structures. The clarity of communication that resulted from this intersection of mathematics and social sciences has been credited with much of the field's early growth (Freeman 1984). As Freeman relates,

[From] the start, contributions to social network analysis were often couched in mathematical terms. The relative precision of these mathematical treatments gave social networks an advantage. Because of that precision, the network field did not generate the same kinds of quibbles and misunderstandings over terms and concepts that lead to conflict in fields that are wedded to a natural language. (2004)

The mathematical core of social network analysis has delivered the dual benefits of precision and flexibility. Equations are clear to the point that those who were interested in the topic could build upon one another's work with minimal need for clarification. But mathematical definitions are also general enough to allow for their application in a variety of relational contexts. The structural measures that form the core of social network analysis have thereby proven to be compelling in a variety of contexts and interests. The logarithmic proliferation in where and how social network analysis has been applied (Otte and Rousseau 2002, Freeman 2011) is testament to the scalability of the field.

The diverse purposing of social network analysis has been mirrored by a corresponding proliferation in the number and variety of software packages that are available today in the field of social network analysis. Although software developers and programmers have put a great deal of effort into producing network analytic software that is suited to a wide variety of

needs and applications, no single piece of software is generally applicable to every situation. Software packages have been optimized for efficiency, analytic variety, analytic specificity, ease of use, specialized data handling, greater capacity for visualization, and for terminology and concepts that are tailored to a particular end-user. Software also differs in style of user interface, method of reporting, and even the default methods for scaling output.

As the available packages continue to diversify, their content is also converging. However, the question of whether the analytic functions across programs are truly equivalent and exchangeable arises. Are the names being used to identify each function explicit in what they identify, or are they only referring to a generalized class of functions?

Naming conventions are important. The developers of network analytic software are faced with decisions about how they should implement a particular analytic function. Some software developers may choose to incorporate the ability to handle common topological features of social networks (e.g., loops, multiple components) by default, while others may choose a stricter interpretation of how the measure or algorithm should perform. Under such a paradigm, the terms in use within the social network analysis community become less precise over time and diverge from the original strength of network analysis: clarity. It is possible that a measure or algorithm differs by implementation in order to address some given scenario or feature of network topology, and that it therefore bears unique attributes that constitute a trade-off at some level. It is therefore valuable to both analysts and the social network analysis community for any such differences to be explicit, or systematized.

The issue of whether two software implementations produce the same measures is especially important when using two programs in concert. In such situations, consistency of output indicates that the user is introducing a minimum of variability when moving from one program to

another. The equivalency of network metrics is important as small variations in such basic measures may translate into large differences on more complex algorithms. However, procedural dissimilarities in programs' calculations of measures can be difficult to identify and frequently lack documentation.

Differences in how various software programs provide output constitute a barrier to assessing consistency of measures from program to program. The variety in default output styles (e.g., raw scores, normalized scores, scalar multiplied output) makes it difficult to visually compare raw output. Even if there were to be no meaningful difference in the *information* provided in the output, the empirical differences that are evident on casual inspection make such a judgment more difficult to establish. The numbers may look different, but in many cases the user would never know just by visual inspection.

ally, the investigation involved the four most commonly applied centrality measures: degree, betweenness, closeness, and eigenvector (Valente et al. 2008). Such measures are often fundamental to social network analysis. Here, we evaluate and report on the consistency of basic measures of node centrality from across various software platforms in standardized simulations.

Materials and Methods

Six software packages for social network analysis were compared in terms of their calculations of four basic measures of node centrality in each of four networks. We selected popular network analytic software that are self-contained (UCINET, Pajek, ORA, and Gephi), or available through CRAN R archive (sna and igraph) packages (*Table 1*, below).

Table 1: Analytic interfaces used in this study

	Version	Source
UCINET	6.564	http://www.analytictech.com/
Pajek	4.03	http://pajek.imfm.si/
ORA-NetScenes	3.0.9.9.20	http://www.casos.cs.cmu.edu/projects/ora/
Gephi	0.8.2	http://gephi.org/
sna	2.3-2	http://www.statnet.org/
igraph	0.7.1	http://igraph.org/

Our primary focus is an assessment of inter-program reliability and three related questions. Are the various software packages producing consistently equivalent results? If not, how do they differ? Under what conditions do the centrality outputs diverge, if divergences exist?

To assess inter-program reliability, we focused on the basic building block of network analysis: node centrality. Specifi-

Measures

The “big four” centrality measures (degree, betweenness, closeness, and eigenvector¹² centrality) were calculated using each ana-

¹² The program Pajek does not include a measure titled eigenvector centrality. In cases of undirected networks, Pajek’s “hubs and authorities” measure is analogous to eigenvector centrality and was used for the purpose of comparison.

lytic interface. These measures were selected because they are basic measures that are frequently used alone for analytic inference, as well as functioning as constituent parts of more complex algorithms. Variability in these measures may lead to downstream variation in analytic results for more complex algorithms. Some programs produce output in multiple formats (e.g., raw, normalized, scaled). When possible, similarly scaled output was compared (Table 2, below).

if present, is a singular anomaly (e.g., differences in floating point) or appear to be deviations that are patterned (i.e., errors that arise from differences in the assumptions behind how a measure should be calculated).

Datasets

Network data (graphs) of variable size and modality were generated to compare centrality measures across software packages. Undirected one-mode [small ($n=35$) and a moderately large ($n=2000$)] and two-mode

Table 2: Output scaling

	Degree	Closeness	Betweenness	Eigenvector
Gephi	Raw	Normalized Average	Raw	Scaled (max=1)
Pajek	Raw	Normalized	Normalized	Normalized
UCINET	Raw	Average	Raw	Normalized
ORA	Normalized	Normalized	Normalized	Normalized
sna	Raw	Normalized	Raw	Normalized
igraph	Raw	Normalized	Raw	Scaled (max=1)

Centrality measures were deemed to be optimally *correspondent* if the Pearson correlation coefficient comparing two centrality values was 1.0000. The closer the correlation is to the optimally consistent value and when scatterplots lie along a 45° line, the better the centrality values concur between software calculations. Optimal consistency is invariant to scale and magnitude differences in raw centrality values. If the correlation fell below 1.0000, then there was suggestive evidence that these measures lacked consensus across software.

Scatterplots offer additional insight for comparing similar measures that employ different scales. If any nodes are of particular interest to the analyst then potential variations in their measurement may become very important. Deviations in the measurement of a small number of nodes within a large network could still occur within the correlation threshold that we selected. Scatterplots were therefore used to identify or characterize variation in measurement and assess whether such variation,

[small ($n_1=10$, $n_2=25$) and a moderately large ($n_1=300$, $n_2=1500$)] network datasets were generated. Initially, both the one-mode, and two-mode networks contained smaller disconnected components (e.g., isolates and/or other small components) in addition to a large main component (Table 2). One-mode networks also contained loops.

New networks were created by removing loops, removing smaller components, or both, from the initial network in order to model a variety of conditions. This resulted in twelve networks: both large and small networks that contain either loops, or disconnected components, or both, or neither; as appropriate for one- or two-mode networks.

Each dataset was designed to be well within the data handling limits of each of the software packages that we evaluated. Most of the programs tested were limited mainly by concerns such as network density and size, in addition to the processor speed and the amount of available memory in a given computer. All are

Table 3: Data used for reliability comparisons

Data	Nodes in Main Component	Nodes in Smaller Components	Number of Loops	Max. Number of Nodes	Average Degree
<i>Small One-mode</i>	29	6	6	35	3.5
<i>Large One-mode</i>	1876	112	60	2000	3.0
<i>Small Two-mode</i>	(10, 21)	4	NA	(10, 25)	3.7
<i>Large Two-mode</i>	(300, 1815)	185	NA	(300, 1700)	3.7

capable of handling networks into the tens of thousands of nodes, with some capable of handling networks into the millions of nodes.

Centrality measures were calculated using all six programs under a variety of conditions on all four networks, where applicable.¹³ All graphs were undirected, with no multiple edges. These graphs were analyzed under multiple conditions: 1) loops (edges that recursively link a node to itself), but no disconnected components present; 2) disconnected components, but no loops present; 3) loops and disconnected components present; and 4) a reference graph with no loops or disconnected components. For a more detailed description of the procedures used for each software program, see *Appendix 1*.

Results

Our findings, presented in brief form in Table 4, demonstrate that differences between analytic programs exist on each measure, with the notable exception of betweenness centrality. Results are presented below by measure, and within each measure, by network condition. Results are presented in a manner that highlights some of the most common or notable differences between programs. Consistency was considered to be “high” when no notable

difference arose, “medium” when the output from *one or two* software implementations differed from others, and “low” when the output from *more than two* software implementations differed from others. For the sake of brevity, many of the cases where all programs demonstrated high consistency in the measures produced (“High” in Table 4) are not discussed but may be noted in the table below. For all measures, with the exception of eigenvector, differences in output were generally more pronounced in smaller networks.

¹³ Loops were not considered to be a feature that is consistent with the definition of two-mode networks as consisting of two sets of nodes that have ties between but not within each node set. The two-mode networks were therefore not evaluated for network data with loops.

Table 4: Consistency of output by centrality type and network conditions

	No Disconnected Components & No Loops		Disconnected Components		Loops		Disconnected Components & Loops	
	1 Mode	2 Mode	1 Mode	2 Mode	1 Mode	2 Mode	1 Mode	2 Mode
Betweenness Centrality	High	High	High	High	High	NA	High	NA
Degree Centrality	High	Medium	High	Medium	Low	NA	Low	NA
Eigenvector Centrality	Medium	Medium	Medium	Low	Medium	NA	Low	NA
Closeness Centrality	Medium	Medium	Low	Low	Medium	NA	Low	NA

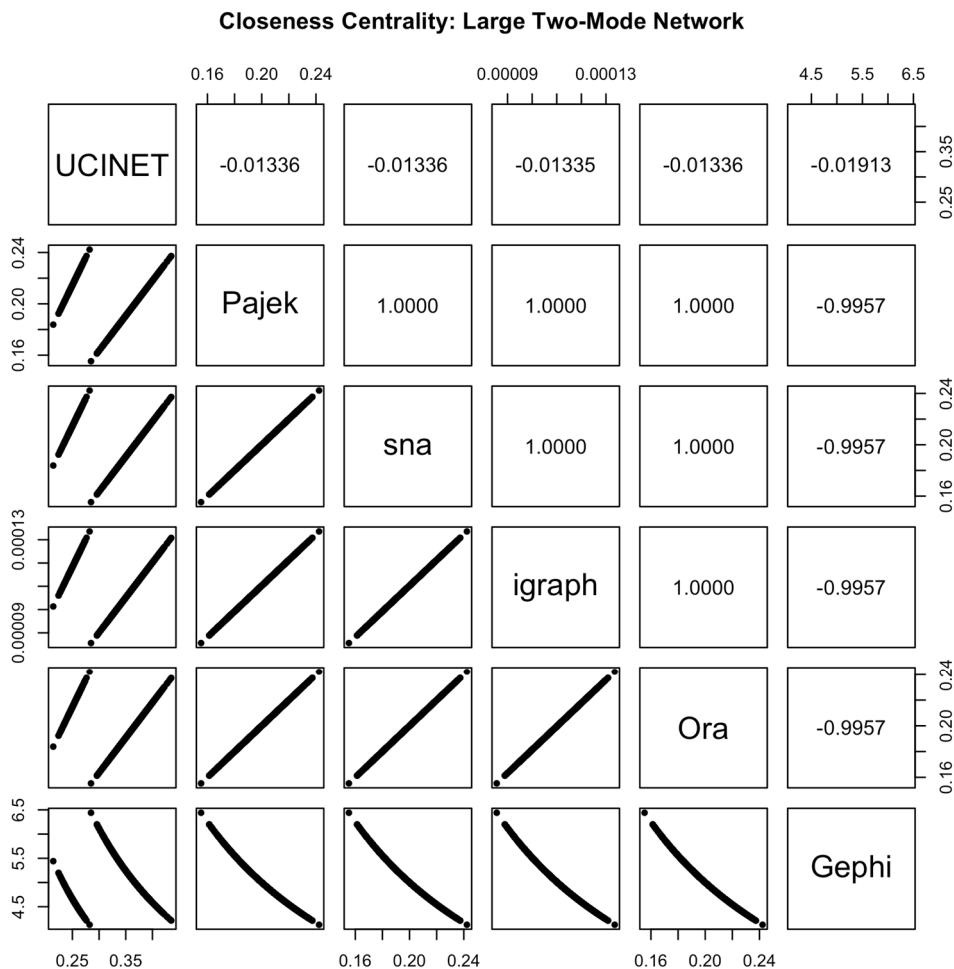
High = Completely consistent, *Medium* = One or two programs vary from the others, *Low* = More than two programs offer unique results

Closeness Centrality

Closeness centrality measures showed the least amount of measurement variability in ideal networks (i.e., no loops or disconnected components), or in networks containing loops, but not in disconnected components. In networks containing *no loops or disconnected components*, plots indicated that calculations of closeness centrality were consistent between Pajek, sna, igraph, ORA, and UCINET; but only when UCINET was calculated using Freeman (1979) normalization.

UCINET and Gephi also correspond when closeness measures in UCINET are reported as summed or averaged distances. In this condition, both UCINET and Gephi produce output for Freeman closeness with smaller values indicating shorter average distances from a particular node to all others in the graph (see negative correlation coefficients [small graph $r = -0.9903$, large graph $r = 0.9883$]). UCINET also offers an “Average Reciprocal Distance” measure (ARD) that corresponds more closely with other programs (small graph $r = 0.9850$, large graph $r = 0.9990$).

Figure 1: Scatterplot matrix comparing closeness centrality output for a large, two-mode network. Pearson’s correlation coefficients between programs are provided above the diagonal.



In two-mode networks, neither UCINET, nor Gephi produced results that corresponded with other programs (Figure 1). UCINET, the only one of these programs to include a closeness measure designed explicitly for use with two-mode data, produced a bifurcated plot in both large and small two-mode networks, though the effect is more pronounced in the larger networks (pictured, Figure 3). While the numeric values are different than those seen in degree centrality, the split-line pattern was similar to that observed in two-mode data without loops, but with disconnected components included, and is attributable to UCINET’s distinctive treatment of two-mode output.

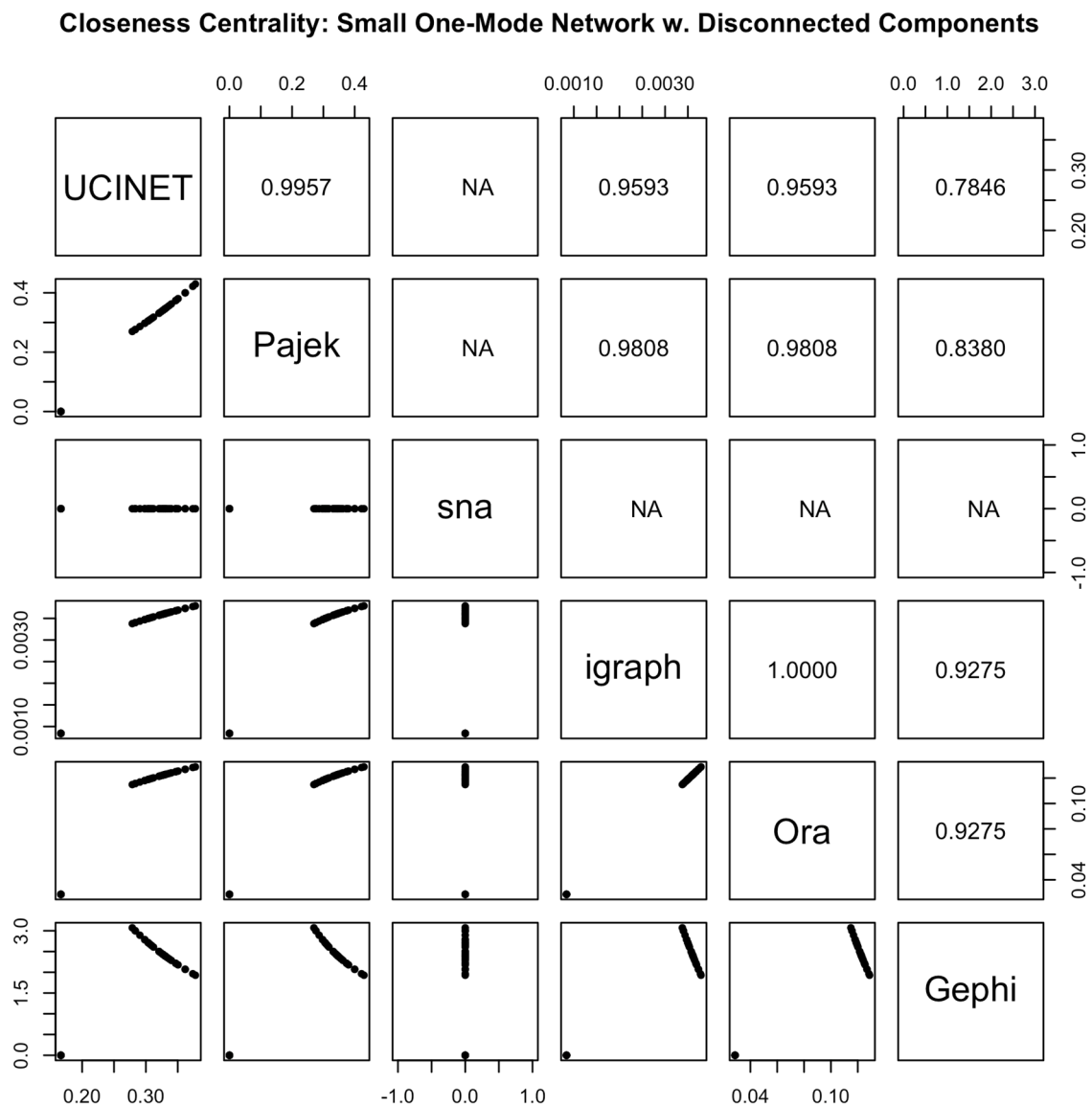
Networks that contain *disconnected components*, but no loops, resulted in the greatest disparities in closeness centrality measurements. Although all software tested cited Freeman as the reference for their centrality measure, only sna seemed to closely implement Freeman’s (1979) approach, and therefore produced no centrality values when disconnected components were included in the graph, as expected under Freeman’s approach. All other tested software generated closeness centrality values, as did sna when disconnected components were not present. Although UCINET (as of version 6.452) no longer provides a warning to the user that analyzing a disconnected graph with Freeman’s close-

ness centrality measure is technically inappropriate, it does require the user to select between options for handling the undefined distances offered by disconnected components. Of the software that produced output under these conditions, values were disparate and only igraph and ORA were consistent with one another. (Figure 2)

In considering networks with *both loops and disconnected components*, there

was a similar disparity of closeness centrality measures as seen in graphs with no loops, but with disconnected components. The same pairs of consistent and inconsistent software values as seen in the graphs with disconnected components were observed with loops added to the data.

Figure 2: Scatterplot matrix comparing output for closeness centrality in a *small, one-mode* network. Pearson’s correlation coefficients between programs are provided above the diagonal. Note that the sna package for R does not produce measures between disconnected components, resulting in correlation values listed as “NA”.

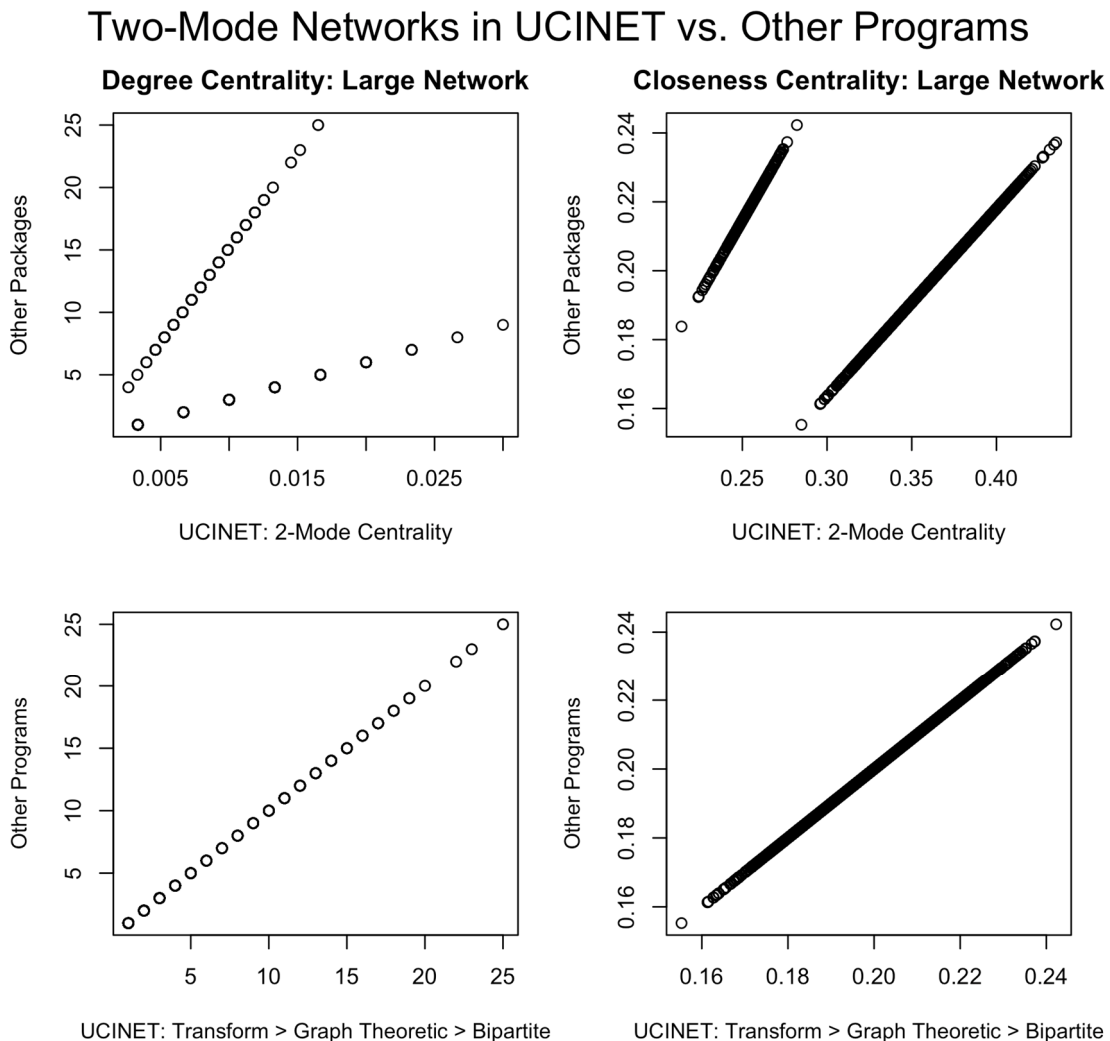


Degree Centrality

In networks with *no loops*, degree centrality was consistent across software in one-mode networks, with the exception of UCINET. A similar pattern was observed for two-mode networks. For these two-mode data, UCINET values fell into two distinct groups in the plots contrasting UCINET with other software. The data are positively correlated, but some stratification is present. This pattern was similar for both large and small two-mode networks, though the effect is more pronounced in the larger networks (Figure 3). UCINET

normalizes output for nodes in each mode individually, an aspect that differentiates it from other tested programs when handling two-mode data. Such differences between UCINET and other programs are eliminated if the network is converted to bipartite, to be analyzed as a one-mode network. The measurement of degree centrality was consistent among all programs in networks that contained *disconnected components with no loops*.

Figure 3: A variety of solutions are possible when analyzing two-mode networks in UCINET. Top Row: Scatterplots of UCINET’s degree ($r = 0.3713$) and closeness ($r = -0.0134$) output using the two-mode centrality procedure, compared with other analytic packages. All other packages performed identically. Bottom Row: When transformed into a bipartite network format, UCINET calculates as for a one-mode network, and results are analogous to other packages. Closeness centrality for the bipartite aspect was calculated using Freeman normalization in UCINET.

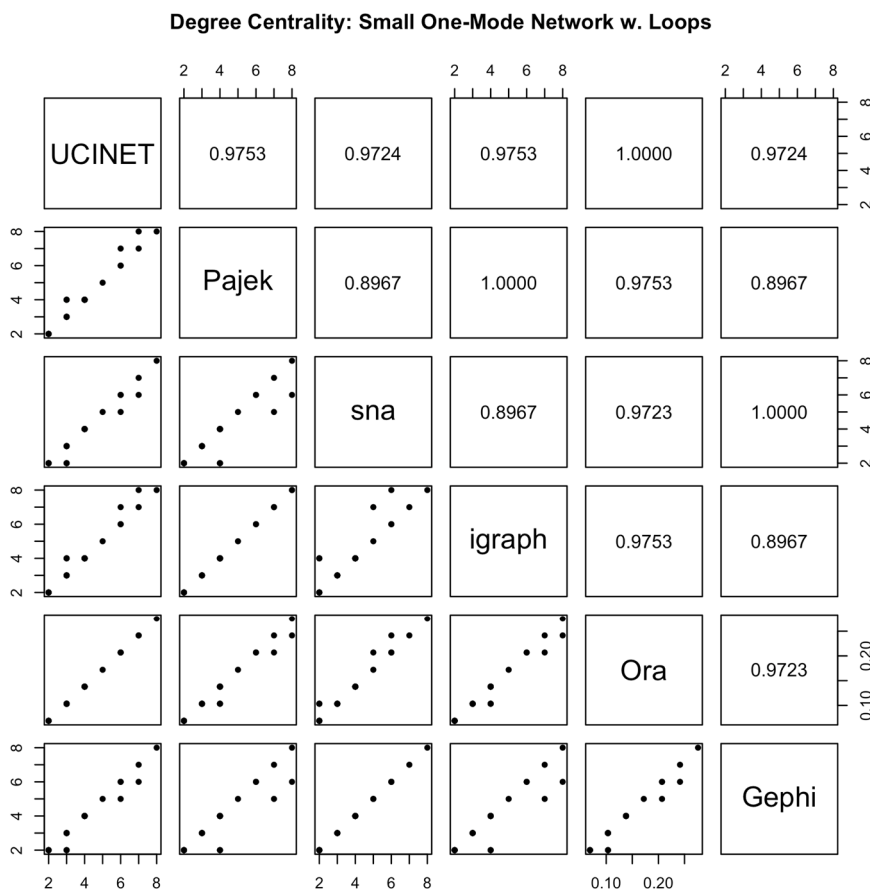


One-mode networks containing loops generated the greatest variability in measures of degree centrality across software (*Figure 4*). For both small and large networks with loops, calculations of degree that are made without modification of the data structure were consistent only between UCINET and ORA, between igraph and Pajek, and between sna and Gephi (see *Figure 4* for an example in a small network). In networks with *both loops and disconnected components*, the patterns were essentially the same as those observed for one-mode networks with loops only. No other new patterns were apparent.

The variations in output stem from how each program handles loops. Program

defaults counted single loops as two edges (Pajek and igraph), loops as one edge (UCINET and ORA), or ignored loops entirely under the default commands (Gephi and sna). Note that the two R packages (sna and igraph) differ in their default treatment of loops, with igraph defaulting to include loops and sna defaulting to ignore them in calculations. When the sna package was modified to include loops (`diag = TRUE`) in the calculation of degree centrality, sna counted all loops as one edge and the output was consistent with that of UCINET and ORA. Gephi counted all loops as two arcs in a manner that was consistent with Pajek and igraph.

Figure 4: Scatterplot matrix comparing degree centrality output for a small, one-mode network containing loops. Pearson's correlation coefficients between programs are provided above the diagonal.

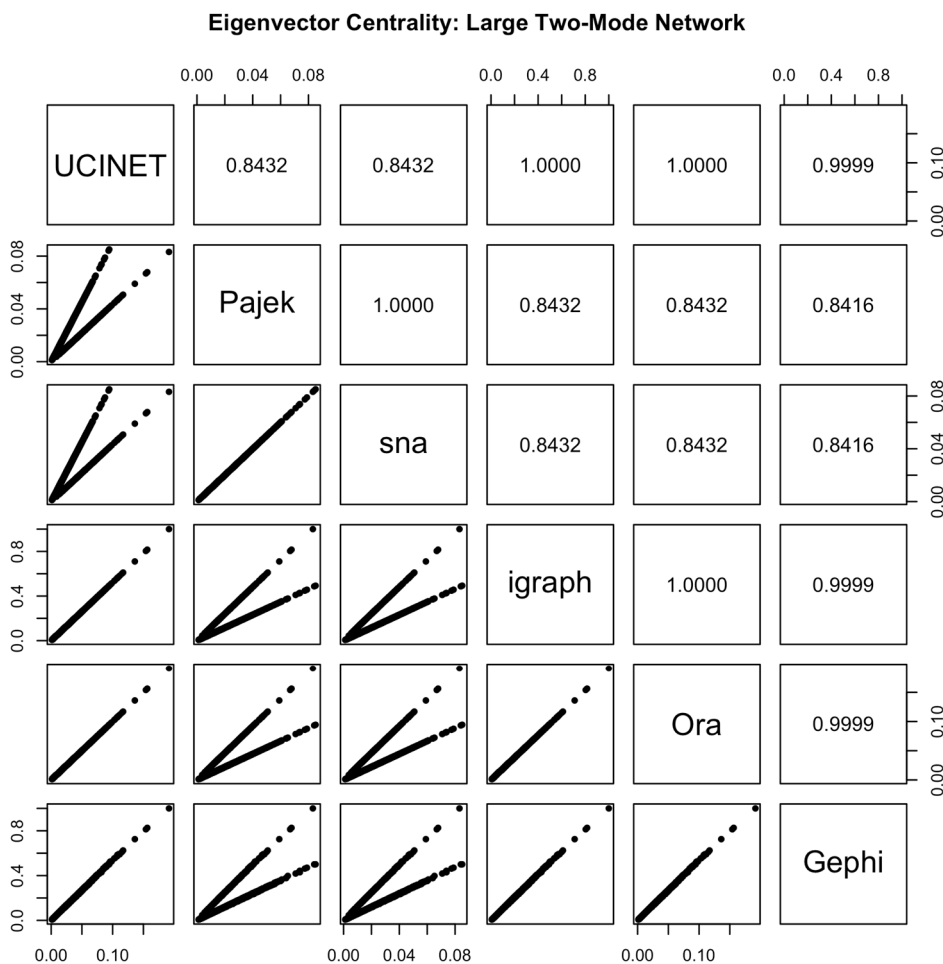


Eigenvector Centrality

Eigenvector centrality was inconsistent across software packages and network types. In networks with *no loops or disconnected components*, eigenvector centrality measures were inconsistent between Gephi and other programs in moderately large, but not small one-mode networks. Changing the default number of iterations in Gephi’s eigenvector centrality measure from 100 to 1,000,000 greatly improved the consistency of measures between programs; however, a small disparity remains for one-mode networks ($r = 0.9901$).

In two-mode networks, igraph, ORA, Gephi, and Pajek’s “2-mode important vertices” function produced results that were largely consistent with UCINET’s two-mode eigenvector centrality (*Figure 5*). Pajek’s “hubs and authorities” measure (designed for one-mode networks) and sna produced results that are consistent with one another (not shown). In large two-mode networks, however, the output from Gephi was again characterized by some small disparities (*Figure 5*).

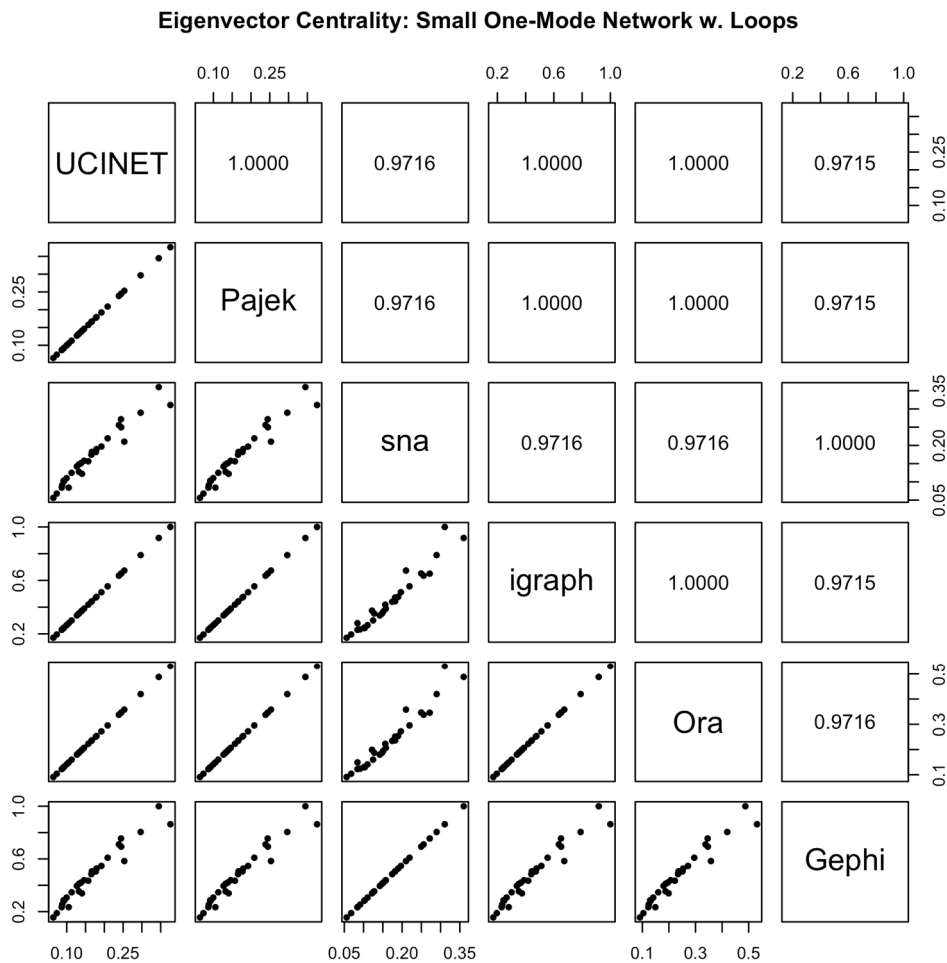
Figure 5: Scatterplot matrix comparing eigenvector centrality output for a *large, two-mode network*. Pearson’s correlation coefficients between programs are provided above the diagonal. Pajek output for this plot was calculated using “important vertices”, a two-mode generalization of hubs and authorities.



Networks containing loops, but lacking disconnected components resulted in additional variability in measures of eigenvector centrality across software. As observed for degree centrality, the correlation between programs' centrality values was high; however, a separate set of points forming a group off of the diagonal appeared. Eigenvector centralities calculated in large, one-mode networks resulted in correspondence between UCINET, ORA, igraph, and the "hubs and authorities" measure in Pajek. The result was similar in small one-mode networks. A correspondence between sna (which defaults to ignoring loops) and Gephi is also noted (Figure 6). The sna "evcent" function offers two

additional options for calculating eigenvector centrality. The sna evcent function with included loops (diag=TRUE argument) yielded eigenvector scores correlated ($r = 1.0$) with all other software except for Gephi results. However, when combining presence of loops with the more robust calculation of eigenvector centrality (diag=TRUE, use.eigen=TRUE) specified in the user manual, the outputted eigenvector is inversely correlated (sna : other packages, $r = -1.0$; sna : Gephi, $r = -0.89$). The variability in eigenvector centrality scores was noted in large and small networks, but more pronounced in the former.

Figure 6: Scatterplot matrix of eigenvector centrality output for a small, one-mode network with loops. Pearson's correlation coefficients between programs are provided above the diagonal. Note, initial calculations in sna – shown above – were run using the default argument (diag=FALSE). For additional variation, consult the text above.

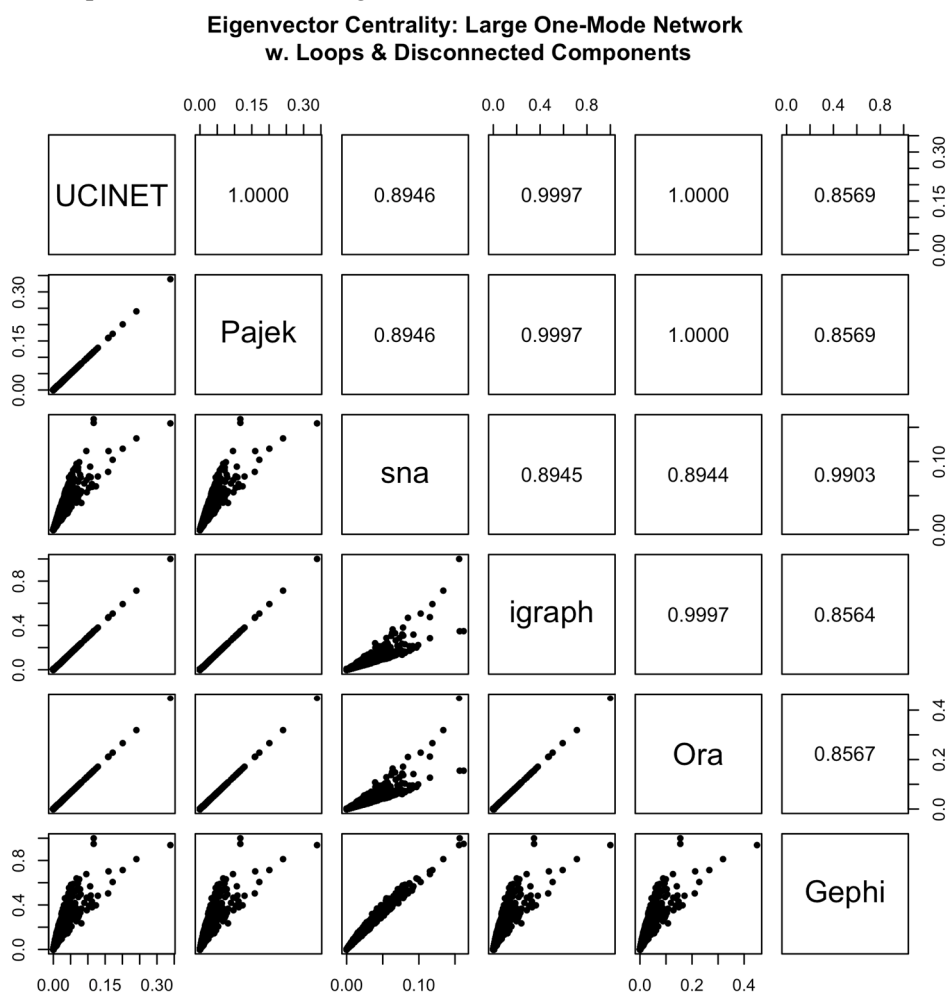


The igraph package produced eigenvector output that differed slightly from other programs in networks that contain *disconnected components, but no loops* (small networks $r = 0.9890$). The disparity was much less pronounced in large networks ($r = 0.9997$).

The above patterns of inconsistencies in calculating eigenvector centrality persisted in networks with *both loops and disconnected components*. In small networks of this type, Pajek, UCINET, and

ORA produced measures that were consistent with one another. Similarly, sna and Gephi also produced nearly identical output in small networks with both loops and disconnected components. In larger networks, however, the similarities between sna and Gephi diminished. Only Pajek, UCINET, and ORA were highly consistent (see *Figure 7*).

Figure 7: Scatterplot matrix comparing eigenvector centrality output for a moderately large, one-mode network containing loops and disconnected components. Pearson’s correlation coefficients between programs are provided above the diagonal.



Betweenness Centrality

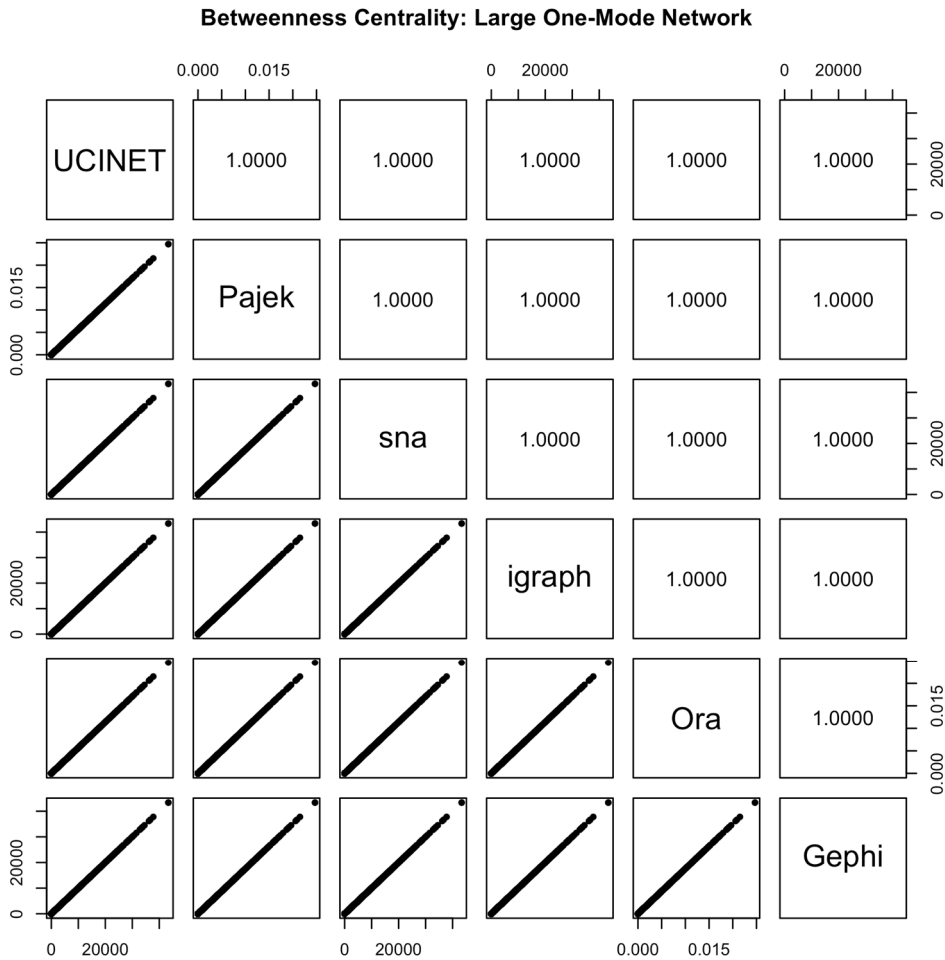
Measurement of *betweenness centrality* was virtually unaffected by the various network conditions being evaluated (i.e., loops, disconnected components). Measures were consistent for each of the

tested packages, on every dataset (see *Figure 8* for an example). The one, very slight, exception was in UCINET’s two-mode measures. UCINET differed slightly from other programs in measuring betweenness in the small two-mode network

($r = 0.9996$, accompanied by slight jitter in scatterplots – not shown). However, no differences were apparent if the same network was converted to bipartite and the one-mode variation of the betweenness measure was used instead. For more examples

of the differences between UCINET’s two-mode measures and other programs’ approaches to two-mode networks, see *Figure 3*.

Figure 8: Scatterplot matrix comparing betweenness centrality output for a *large, one-mode* network. Pearson’s correlation coefficients between programs are provided above the diagonal.



Discussion

This study was designed to examine basic reliability concerns among a selection of popular tools in use within the social network analysis community. Specifically, we investigated whether some popular software packages were producing equivalent results. We found variability that brings to light disagreement, sometimes substantial, over how four concepts of node centrality should be measured. The programs under consideration were only able to produce the same output under a very narrow set of conditions. Disagreements over aspects of how these measures should be operationalized manifested as networks departed from the ideal reference graphs that contained no loops or disconnected components. Such variability precludes the ability to seamlessly port data and/or exchange measures between programs and makes it essential for the user to have access to evaluations that highlight differences between the default, and available, options for various measures when using two or more programs in concert. Within the social network analysis community, the differing assumptions behind the various measurement variations unnecessarily cloud communication between users of different programs and leave enough doubt in the minds of new entrants as to whether the community has unified its language. Below, we discuss in greater detail our interpretation of the results, the implications of our findings for the average user, and the implications for the social network analysis community.

General findings

By employing hierarchical subsets of network conditions, we isolated measure dif-

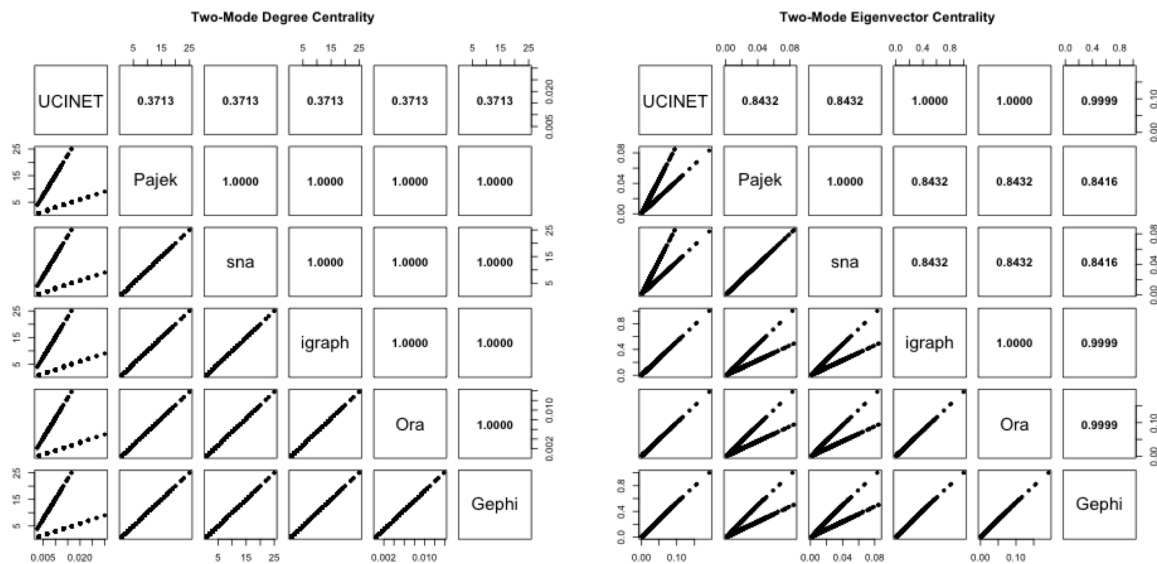
ferences under specific conditions. The use of varying network conditions was intended to better reflect a range of network data that are likely to be encountered. Conditions in the undirected networks ranged from the “ideal” of reference data – no loops or disconnected components – to scenarios commonly encountered when analyzing social networks, namely, loops, disconnected components, and the combination of the two.

In general, centrality measures for reference graphs – those with no loops or disconnected components – were largely consistent. This may be taken to imply that programs are implementing the same – or very similar – algorithms for the offered measures, albeit mainly in the absence of issues that may complicate calculation, such as loops and disconnected components.

A notable inconsistency did, however, arise in the analysis of the two-mode reference networks. Of the tested software, only UCINET offered measures tailored specifically for two-mode networks. Correspondingly, analytic results reveal a bifurcated pattern when comparing calculations of degree and closeness in UCINET to those of other software packages, along with slight differences in betweenness measures in small two-mode networks. No other programs demonstrated a pattern that corresponded to that of UCINET when measuring degree, closeness, or betweenness. When measuring eigenvector centrality, however, three additional programs (igraph, ORA, and Gephi) evince a bifurcated pattern that corresponds very closely with UCINET. (*Figure 9*)

Figure 9: Scatterplot matrices comparing degree and eigenvector output for a *two-mode* network. Neither network contains loops or disconnected components. Pearson's correlation coefficients between programs are provided above the diagonal.

Variability in Handling Two-Mode Networks



There was a surprising amount of inconsistency in the most basic measure: *degree centrality*. Programs' definitions identified degree centrality as either the number of neighbors adjacent to a node, or the number of edges incident upon a node. However, many programs did not provide a citation for this measure. Among those that do, the Freeman (1979) definition is employed, which does not account for a topological feature that is common in biological, corporate, citation, and other networks: self-referencing, or loops. Freeman implements a variation on Nieminen (1974), which accounts for the number or proportion of other nodes that are adjacent to a particular node, but not for nodes that are adjacent to themselves (i.e., loops). The evaluated programs defaulted to three different methods for dealing with loops in a graph, revealing the variation in degree centrality calculations. This disparity between definitions of what constitutes a loop and how its effect should be measured in such a conceptually simple calculation suggests the

need for the community of social network analysts to strengthen naming and measurement standards. Such a process may reduce error in interpretations resulting from what are actually different measures residing under the same name.

The problem of different measures residing under the same name is exemplified when considering *eigenvector centrality*. Although most programs identify this measure as eigenvector centrality, the presence of loops in a network reveals slight differences in how this measure is operationalized. In its essence, eigenvector centrality extends degree centrality by weighting each node's score by its neighbors' scores (Bonacich 1972). Like degree, it is affected by loops. Five of the tested programs cite Bonacich (1972, or 1987) in calculations of this measure, and one – Pajek – employs the analogous “hubs and authorities” (Kleinberg 1999). As with degree centrality, the presence of loops in a network reveals which programs operationalized this measure using the same or substantially similar assumptions.

Three programs – UCINET, ORA, and Pajek – were consistent with one another in measuring eigenvector centrality in all three variations of one-mode networks. This is notable because one of those three, Pajek, provides “hubs and authorities”, which generates two independently scaled measurement vectors, of which the authorities vector was consistent with eigenvector centrality in the other two programs. This is the one case where a centrality measurement that differed from the classic citation was identified under a different name.

Perhaps the most conspicuous case of inconsistent calculations is *closeness centrality* in the presence of disconnected components. All programs evaluated referenced Freeman’s (1979) measure, with the exception of Pajek, which cites Sabidussi (1966), as cited in Freeman. However, it quickly becomes apparent that the question of how to operationalize closeness centrality is neither agreed upon, nor settled in consideration of disconnected components. The original formula for closeness centrality should not function with disconnected data since the distance between disconnected components is undefined (Freeman 1979). Any means of dealing with disconnected data, with the possible exception of running calculations only *within* each individual component, is therefore a later variation of the Freeman formulation. Isolates and other disconnected components – which can be common network features in some areas – frequently present an obstacle to communicating the results of this measure. A wide array of alternate measures has since been proposed to allow calculation of closeness with disconnected data (e.g., Borgatti 2006, Dangalchev 2006, Opsahl 2010, Wei et al 2011). Amid such proliferation, however, it is unclear which forms have been incorporated in the software yielding results for disconnected component datasets.

Only the R package sna produced an error message without numerical results rather than closeness values as stipulated in Freeman (1979), because it treats the distances between disconnected components

as infinite. There is also a stern admonishment in the package details against calculating closeness centrality in networks with disconnected components. All other software produced closeness calculations without requiring that smaller components first be removed.

The analysis of disconnected networks using closeness produced widely varied output. Correspondingly, all tested software provided some means for dealing with disconnected components. In most software, the method for defining the distance between disconnected components was incorporated into the measure. ORA and the R package igraph appear to default to substituting the number of nodes for undefined distances, whereas Gephi appears to report undefined distances as zero and omit disconnected nodes from calculation. Pajek sets undefined distances to zero and calculates closeness only within each component.

Of the software tested, UCINET offers the most options for applying closeness centrality to one-mode networks with disconnected components. The user may choose one of four options for dealing with the distances between disconnected components: (1) substitute the number of nodes in the graph for the undefined distance; (2) substitute the maximum distance, plus one (the default setting); (3) treat undefined distances as missing and assign no value to isolates; and (4) set undefined distances to zero and calculate closeness only within each component. Those four options, combined with three options for scaling output (summed distances, averaged distances, and Freeman normalization) present the user with 10 combinations of options¹⁴ for calculating closeness in cases with disconnected components.

Aside from differences in how output was scaled, there was essentially no variation in calculating *betweenness centrality*. This is perhaps unsurprising, as there is little room for interpretation in the definition of this measure. Betweenness, a

¹⁴ The option of treating undefined distances as missing is scaled in only one manner.

normalized count of the number of times that a node appears on the shortest path between any two other nodes (Freeman 1977), will generally be unaffected by disconnected components and loops since loops will not create new geodesics (shortest paths) and the absence of paths between disconnected components does not complicate geodesic counts.

It bears repeating that the programs tested displayed relatively little variation when analyzing reference graphs – those with neither loops, nor disconnected components. The variation that was present in the centrality output from the reference graphs appears more likely to have arisen from differences in opinion on preferred methods of calculation in situations that vary from the ideal of connected one-mode graphs with non-recursive edges.

Implications for the field user

The reference graphs make it clear that – with only a few exceptions – those responsible for developing and maintaining each of these programs have done an admirable job of benchmarking their programs against others and correcting unintentional software differences. However, network topology that diverges away from the “ideal” reference graph reveals that there is a great deal of disparity in the analytic assumptions that are built into software used to calculate such measures. The problem that arises from this lack of understanding and agreement within the social network analysis community is that it puts both analysts and the field itself at a disadvantage by introducing unnecessary noise into analyses and communication within the community.

Certainly, for those analysts whose data are similar to our reference graphs (i.e., no loops, no isolates or other disconnected components) the low variability in measurement definition and implementation is good news. The lack of variation in the output for the four reference graphs indicates that the programs used in this study agree in their standards for the calculation of basic centrality measures under the most

basic and favorable conditions. The differences that resulted from other conditions, however, underscore the importance of an analyst’s familiarity with their choice of software, and the software used by those whose work they wish to use as an analytic benchmark. A good deal of care should be exercised to verify the precise method of calculation being applied and the settings – and defaults – that were employed for those calculations.

Centrality measures typically form the foundation of an analysis and if their implementation varies, more complex algorithms that involve one or more of these centrality measures may be producing measures or results that magnify this variability. Unfortunately, the variation in the measurement of centrality values *between* programs remains somewhat opaque. Measurement disparities were observed even when the terms and citations used to identify the measures were identical between programs.

Clearly, analyzing the same network in the same way, using different software, can produce divergent results. If the implementation of a given centrality measure differs from one program to another, they are at best two *different* variants of the same measure. If such is the case, it will aid the analyst to know which variant they are utilizing. With its variety of measures and selections, UCINET goes the furthest of all the software tested above in identifying which variant of a particular measure it is employing – naming particular variants of the same measure according to its originator or an intuitive description of its function. Only a few of the tested analytic tools were consistently explicit about the *equations* used to produce all four measures.

The clarity in communication that has characterized the development of methods in the field of social network analysis is less evident in software operationalization. This presents a threat to the validity of how those measures are employed. Definitional differences between programs exist and are not readily apparent

to the average user. Although variations in centrality calculations hold the potential to increase the validity of a particular measure when applied in the appropriate context, such differences are frequently masked from the general user, resulting in increased potential for the misapplication of measures.

The present research has highlighted the value of knowing what one is getting into when considering new analytic software, and the importance of thoroughly vetting the topology of the network being analyzed. Add to that the tendency for most software packages to have some provision for porting data and/or measures between packages. The detected disparity in measures available in the current analytic packages indicates that such practices should be undertaken with caution – especially in cases where graphs contain loops, isolates, or disconnected components. If the basic measures differ between packages then it may be inadvisable to use the two packages together in an analysis that involves those measures.

Implications for the social network analysis community

The lack of consensus over how to operationalize the most common node centrality measures suggests some ontological variability within the social network analysis community. Disagreements over how various centrality measures should be operationalized would not be troubling if they were apparent to the user. But the differences highlighted above are far from clear to the average user. Lack of agreement over how to operationalize a measure is masked when a variety of approaches share a single name. The situation is exacerbated when software documentation does not clarify precisely which approach has been implemented.

The debate over how various network measures should be calculated is rich, and as old as the field itself. The community's openness to new variations on established methods provides flexibility and a healthy diversity of analytic options.

However, the advantages of such wealth are substantially diminished when the same measure is operationalized differently in each analytic package. Although a shared lexicon of terms and concepts exists within the social network analysis community, those terms and concepts are only generally – and not explicitly – applied. The programs used to perform social network analysis are disparate enough to create idiosyncratic analytic results.

The interfaces of each analytic package vary greatly and do not always default to the most commonly used variants of each measure. Without equivalence of measurement assumptions and nomenclature between programs that is easily accessible, the assumption of equivalence and portability of centrality measures in network analysis is lacking. This increased variability of centrality results may potentially affect more complex algorithms that incorporate these basic measures. These basic differences could be resolved with agreed-upon defaults and naming conventions for variants of a particular measure or algorithm.

It is important that the variants of each measure be identified as distinct variations of a centrality – or other measurement – theme. It is not enough to identify a measure generally as “closeness centrality” if it varies from the basic measure identified by Freeman (or Sabidussi) – which most all do. Instead, the measure should be explicitly identified as a particular variant in order to better emphasize its unique attributes and trade-offs.

Explicit descriptions of measurements can be essential to proper analysis.

To draw an example from another analytic field: *post-hoc* tests for pairwise comparisons of means following analysis of variance have been developed to address variations in hypothesis testing, trade-offs between power and error, unequal sample sizes, and unequal variances (for a discussion of 22 *post-hoc* tests see Kirk 1995). Although the more casual user may find such a selection daunting, the strength in

this diversity of options is that the user may better consider and tailor their analytic selections. Additionally – and perhaps more importantly – small differences between variations on a measure become a feature, rather than an obstacle, when a particular form of a measurement is explicitly named.

Naming conventions are important. If a measure or algorithm differs from others in order to address some given scenario or feature of network topology, then it bears unique attributes that more often than not constitute a trade-off at some level. It is far better for both the analyst and the community for these differences to become less opaque. Of the tested programs, UCINET appears to have gone the furthest in giving attribution to the different variants of each measure. Though, both R packages benefit from the explicit nature of specifying a measurement. This aids the analyst by further clarifying differences between analytic approaches. The community is aided in establishing reliability of measurements between programs; as such clarity makes it much easier to directly compare results from different programs.

It is not necessary for each program to offer every available option – though several have clearly taken steps in that direction. It is likely to be much more helpful to the social network analysis community at large if the measures and algorithms that a program offers are fully identified for appropriate application of their properties. Proper identification will simplify discourse and improve the communication of methods. Such improvements in communication within the field also translate into increases in measurement validity when a measure is identified as a specific variant, rather than just belonging to a general category or class of measures.

Lastly, it should be noted that a lack of agreement from within the community on something as fundamental as naming conventions hints at an arbitrariness that is surprising given the care and rigor of those who have established and expanded the field of social network analysis.

Freeman (1984, 2004) has repeatedly made a compelling case for clarity and precision in communication – as facilitated by mathematical notation – being the factor that set social network analysis apart from similar fields that rely more on natural language for clarification. The benefits of such precision are, however, often frustratingly beyond the reach of those using social network analysis software.

Further, as researchers from other fields continue to adapt and adopt social network analytic methods, the use of standard, specific terms on the available analytic options provides a clarity that aids newcomers to social network analysis in seeing how their field can benefit by adopting a network analytic approach. But the converse is not the case: imprecise definitions need not constitute an invitation for some within the “hard sciences” to forego established network analysis methods in favor of feigning to invent them for themselves. Although co-option will likely continue, there has been an increase in the number of new entrants to social network analysis who give proper attribution (Freeman 2011). Clarity of communication will reinforce social network analysis as a mature and growing field, and de-emphasize the perception of it as being a general perspective or a mere category of tools (Knoke and Yang 2008, Snowden 2005).

In most cases, it is possible to force the centrality output for a network containing loops or disconnected components to be relatively consistent across all six platforms employed above. But such actions frequently require transformations or other preprocessing in order to do so, and those steps are seldom stipulated since there is no real agreed-upon *definition* of exactly which mathematical approach constitutes each type of centrality. The clarity that comes with definitional consistency *between programs* is what we feel to be needed. We advocate the clarity that comes with dissimilar means to a particular end being clearly identified up front.

The “correct” measure is the one that is best suited to handle the idiosyncrasies of the data an analyst holds. For the analyst to make this assessment, they first need to know the topology of the network they are analyzing; and next, specifically how a measure is meant to operate, and its underlying assumptions. A more complete approach includes asking which variation of the measure is available, the strengths and limitations of that version, and how reliably one or more programs produce accurate measures. We have identified program and inter-program reliability issues under varied conditions. Similar comparisons between other programs and under different conditions are strongly recommended when weighing whether to use two or more analytic programs in conjunction with one another. Further evaluations of inter-program reliability will benefit from adding more types of variation: e.g., directed graphs, density variations, clusterability variations. Ongoing research in this topic will continue to be important as new entrants continue to discover the scalability and utility of the tools and concepts of social network analysis for deciphering increasingly diverse networks with complex topological features.

References

- Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Du Croz, J., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D. (1999). *LAPACK Users' Guide*. (3rd ed.). Philadelphia, PA: SIAM.
- Batagelj, V., Mrvar, A. (2003). Pajek – Analysis and Visualization of Large Networks. University of Ljubljana, Slovenia, EU. Available at: <http://pajek.imfm.si/doku.php?id=download>
- Bastian, M., Heymann S., Jacomy, M. (2009). Gephi: an open source software for exploring and manipulating networks. International AAAI Conference on Weblogs and Social Media. Available at: <https://gephi.org>.
- Becker, R.A., Chambers, J.M., Wilks, A.R., (1988). *The New S Language*. Pacific Grove, CA: Brooks/Cole.
- Bonacich, P. (1972). Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology*, 2, 113–120.
- Bonacich, P. (1987). Power and centrality: A family of measures. *American Journal of Sociology*, 92, 1170–1182.
- Borgatti, S.P., Everett, M.G., Freeman, L.C. (2002). Ucinet for Windows: Software for Social Network Analysis. Analytic Technologies, Harvard, MA. Available at: <http://www.analytictech.com/ucinet.htm>
- Borgatti, S.P. (2006). Identifying sets of key players in a network. *Computational, Mathematical and Organizational Theory*, 121, 21–34.
- Butts, C.T. (2007). sna: Tools for Social Network Analysis. Version 2.2. Irvine, CA. Available at: <http://erzuli.ss.uci.edu/R.stuff>
- Carley, K., Reminga, J. (2004). ORA: Organization Risk Analyzer*. Center for Analysis of Social and Organizational Systems, Carnegie Mellon University. Pittsburgh, PA. Available at: <http://www.casos.cs.cmu.edu/projects/ora>
- Csardi, G., Nepusz, T. (2006). The igraph software package for complex network research. *InterJournal*, Cambridge, MA, 1695. Available at: <http://igraph.sourceforge.net>.
- Doreian, P. (2001). Causality in social network analysis. *Sociological Methods Research*, 30(1), 81–114.
- Dangalchev C. (2006). Residual Closeness in Networks. *Physica*, 365:2, 556–564.
- Freeman, L.C. (1977). A set of measures of centrality based on betweenness. *Sociometry*, 40: 35–41.
- Freeman L.C. (1979). Centrality in Social Networks: Conceptual clarification. *Social Networks*, 1: 215–239.
- Freeman, L.C. (1984). Turning a profit from mathematics: the case of social networks. *Journal of Mathematical Sociology*, 10: 343–360.
- Freeman, L.C. (2004). *The Development of Social Network Analysis: A Study in the Sociology of Science*. North Charleston, SC: BookSurge.
- Freeman, L.C. (2011). The Development of Social Network Analysis: with an Emphasis on Recent Events, in *Sage Handbook of Social Network Analysis*. John Scott and Peter Carrington, eds. Thousand Oaks, CA: Sage, 26–39.
- Kirk, R.E. (1995). *Experimental Design: Procedures for the Behavioral Sciences*. (3rd ed.) Pacific Grove, CA: Brooks/Cole.
- Kleinberg, J. 1999. "Authoritative Sources in a Hyperlinked Environment." *Journal of the ACM* 46(5):604-32.
- Knoke, D., Yang, S. (2008). *Social Network Analysis*. (2nd ed.) Los Angeles, CA: Sage.
- Nieminen, J. (1974). On centrality in a graph. *Social Science Research*, 2: 371–378.
- Opsahl, T. (2010, March 20). Closeness centrality in networks with disconnected components. In: Tore Opsahl. Retrieved November 21, 2011, from <http://toreopsahl.com/2010/03/20/closeness-centrality-in-networks-with-disconnected-components/>

- Otte, E, Rousseau, R. (2002). Social network analysis: a powerful strategy, also for the information sciences. *Journal of Information Science*, 28, 441–453.
- Reminga, J., Carley, K. (2003). Measures for ORA (Organization Risk Analyzer*). CASOS Working Papers. Retrieved November 21, 2011, from http://www.casos.cs.cmu.edu/publications/papers/reminga_2003_ora.pdf
- Sabidussi, G. (1966). The centrality index of a graph. *Psychometrika*, 31, 581–603.
- Smith, B.T., Boyle, J.M., Dongarra, J.J., Garbow, B.S., Ikebe, Y., Klema, V., Moler, C.B. (1976). Matrix Eigensystems Routines – EISPACK Guide. *Springer-Verlag Lecture Notes in Computer Science*, 6.
- Snowden, D. (2005). From atomism to networks in social systems. *The Learning Organization*, 12(6), 552–562.
- Stokman, F.N., Doreian, P. (1997). Evolution of social networks: Processes and principles, Patrick Doreian and Frans N. Stokman, eds. *Evolution of Social Networks*. Amsterdam: Gordon and Breach Publishers, 233–250.
- Valente, T.W., Coronges, K., Lakon, C., Costenbader, E. (2008). How correlated are network centrality measures? *Connections*, 28(1), 16–26.
- Wei, W., Pfeffer, J., Reminga, J., Carley, K. (2011). Handling Weighted, Asymmetric, Self-Looped and Disconnected Networks. CASOS Technical Report: CMU-ISR-11-113. Retrieved November 21, 2011, from <http://www.casos.cs.cmu.edu/publications/papers/CMU-ISR-11-113.pdf>
- Wikipedia (2011). Eigenvector Centrality. In: Centrality. Retrieved November 21, 2011, from http://en.wikipedia.org/wiki/Betweenness#Eigenvector_centrality.
- Wilkinson, J.H. (1965). *The Algebraic Eigenvalue Problem*. New York, NY: Oxford University Press.