Mathematical Description of Network Properties/Measures

This supplement provides the details for calculating the network properties used in the article for both binarized and non-binarized networks (where applicable) along with R code for calculating these measures. All measures presented here are based on undirected ties.

Degree Centrality – Binary Networks:

Degree centrality for binary networks is a measure of a node's importance in the network as a whole, defined as the total number of direct connections that a node has (Wasserman and Faust 1994:178-180). This is the simplest measure of centrality for an actor and can be formally represented as:

$$C_D(n_i) = \sum_j x_{ij}$$

where:

$C_D(n_i)$ = Degree centrality of node *i*

x_{ij} = presence/absence of relation between node *i* and node *j* (0 or 1)

Degree Centrality – Weighted Networks:

Degree centrality for weighted networks is defined simply as the sum of the weights of direct connections from a node to all other nodes in a network (e.g., Barrat et al. 2004; Opsahl et al. 2010). This can be formally represented as:

$$C_D(n_i) = \sum_j w_{ij}$$

where:

$C_D(n_i)$ = Degree centrality of node *i*

w_{ij} = weight of relation between node *i* and node *j* (between 0 and 1)

Degree Centralization – Binary and Weighted Networks:

Degree centralization is a graph level measure that summarizes the variability and distribution of node level degree centrality scores across all nodes in a network. Measures of centralization can be defined in a number of different ways, but in general characterize how central a set of nodes are in relation to some theoretical maximum. In the context of this study, we use the method for calculating degree centralization described by Freeman (1979) which works for both binary and weighted networks:

$$C_D = \frac{\sum_{i=1}^{g}\left[C_D(n^*) - C_D(n_i)\right]}{\left[(g-1)(g-2)\right]}$$

where:

C_D = Graph level degree centralization

$C_D(n_i)$ = Degree centrality of node *i* (based on the binary or weighted measure)

$C_D(n^*)$ = Maximum degree centrality value for any node in the graph (based on binary or weighted measure)

g = the total number of actors in a network

The denominator in this equation defines the maximum possible centralization score given the number of nodes in a network based on a star network (a network consisting of a single node connected to every other node, with no ties among these other nodes). This measure ranges from 0 to 1 and represents the proportion of this maximum possible centralization represented by a set of nodes. In the present context of tie weights indicating similarity of sites' ware distributions, such a star may not actually be possible. (depending on the numbers of wares and sites). The star is still a reasonable basis for normalizing the statistic, but the apparent maximum centralization may not be achievable in a given case. Comparisons across data sets that vary in their numbers of wares and sites thus require caution.

Betweenness Centrality – Binary Networks:

Betweenness centrality in a binary network measures the degree to which a node lies on shortest paths between pairs of other nodes in a network. This measure of centrality assumes that a node can obtain importance by being an intermediary on paths that direct flows through that network. Shortest paths, or geodesics, are defined as the minimum number of direct ties that need to be crossed to get from one specific node to another (Wasserman and Faust 1994:189-191). As defined by Freeman (1977), betweenness centrality for node *i* is defined as the proportion of geodesics between pairs of other nodes in the network which pass through node *i*. This measure of centrality can be formally represented as:

$$C_B(n_i) = \sum_{j<k} \frac{g_{jk}(n_i)}{g_{jk}}$$

where:

$C_B(n_i)$ = Betweenness centrality of node i

$g_{jk}(n_i)$ = number of geodesics between nodes j and k passing through node i ($i \neq j$ or k)

g_{jk} = total number of geodesics between nodes j and k present in the network

Betweenness Centrality – Weighted Networks:

As described in the main document, there have been many measures proposed for assessing betweenness centrality in weighted networks. In the context of this study, we use a measure most recently described by Opsahl and others (2010; see also Brandes 2001; Newman 2001). This weighted measure of betweenness centrality is similar to the binary model described above in that it characterizes the degree to which nodes lie on shortest paths between pairs of other nodes. In this case, however, shortest paths are not defined in relation to the number of ties crossed to get from one node to another, but rather in terms of what is referred to as the "path of least resistance" (Opsahl et al. 2010:248; see Dijkstra 1959). In the example used here, the path of least resistance between two nodes follows the strongest connection at each juncture. In this case, the total length of the path in terms of the number of ties crossed is considered trivial in relation to the strength of those ties, though Opsahl and colleagues (2010) also offer a means for taking the length of paths into account (we do not use this extension here). The definition of shortest paths as "paths of least resistance" can be formally represented as:

$$d^w(j,k) = \min\left(\frac{1}{w_{jh}} + \ldots + \frac{1}{w_{hk}}\right)$$

where:

$d^w(j,k)$ = the shortest path (path of least resistance) between node j and node k

w_{ih} = the weight of the connection between the origin at node j and an intermediate node h on a path between starting node j and target node k

Following the definition of shortest paths in terms of the path of least resistance, weighted betweenness centrality is calculated similarly to the binary measure as the proportion of all shortest paths between pairs of nodes in a network that go through the node in question. Using shortest paths defined in terms of the equation above, weighted betweenness centrality can be represented as:

$$C_B(n_i) = \sum_{j<k} \frac{d_{jk}^{w}(n_i)}{d_{jk}^{w}}$$

where:
$C_B(n_i)$ = Weighted betweenness centrality of node i
$d_{jk}^{w}(n_i)$ = the number of weighted shortest paths that go through node i ($i \neq j$ or k)
d_{jk}^{w} = the total number of weighted shortest paths in the network

Betweenness Centralization – Binary and Weighted Networks:
Like degree centralization, betweenness centralization for binary and weighted networks can be calculated using the same basic equation defined by Freeman (1979). This measure again ranges from 0 to 1 and provides an assessment of how central a set of nodes are in relation to the theoretical maximum. Freeman (1979) demonstrates that the maximum centralization score for a network (denominator) in terms of betweenness centrality is $2/[(g-1)^2(g-2)]$. Thus, betweenness centralization can be formally represented as:

$$C_B = \frac{2\sum_{i=1}^{g}\left[C_B(n^*) - C_B(n_i)\right]}{\left[(g-1)^2(g-2)\right]}$$

where:
C_B = Graph level betweenness centralization
$C_B(n_i)$ = Betweenness centrality of node i (based on the binary or weighted measure)
$C_B(n^*)$ = Maximum betweenness centrality value for any node (based on binary or weighted measure)
g = the total number of actors in a network

As above, the maximum centralization assumed in this measure may not be possible in the site similarity context depending on the total number of ceramic wares and sites.

Eigenvector Centrality – Binary and Weighted Networks:
We discuss eigenvector centrality for both binary and weighted networks together, as they are calculated using the same basic equation. Eigenvector centrality is a measure of a node's importance in a network defined in relation to the first eigenvector of the adjacency matrix of nodes for both binary and weighted networks. A detailed discussion of the relationship between eigenvectors and corresponding adjacency matrices is beyond the scope of this paper, but is described thoroughly by Bonacich and Lloyd (2001). For both binary and weighted networks, a node's eigenvector centrality score is proportional to the summed scores of other nodes to which it is connected (see Bonacich 1972). In other words, eigenvector centrality for a node will increase if a node is either connected to lots of other nodes, or if a node is connected to highly central nodes. For a given adjacency matrix (A), this measure can be formally represented as:

$$\lambda x_i = \sum_{j=1}^{g} a_{ij} x_j$$

where:
x_i, x_j = the eigenvector centrality scores for node *i* and *j* respectively
λ = the first eigenvalue of the adjacency matrix (*A*)
a_{ij} = the value of the relation between node *i* and node *j* in adjacency matrix *A*

As this equation illustrates, eigenvector centrality is calculated the same way for both binary and weighted networks by defining a_{ij} as the presence or absence of a tie between nodes *i* and *j* in the case of binary networks, or the weight of the relationship between *i* and *j* in weighted networks.

Eigenvector Centralization – Binary and Weighted Networks:
Once again, eigenvector centralization can be calculated for both binary and weighted networks using the basic centralization equation provided by Freeman (1979):

$$C_E = \frac{\sum_{i=1}^{g}\left[C_E\left(n^*\right) - C_E\left(n_i\right)\right]}{\left[(g-1)(a-b)\right]}$$

$$a = \sqrt{\frac{g}{2}} \qquad b = \sqrt{\frac{g}{\left[2(g-1)\right]}}$$

where:
C_E = Graph level eigenvector centralization
$C_E(n_i)$ = Eigenvector centrality of node *i* (based on the binary or weighted measure)
$C_E(n^*)$ = Maximum eigenvector centrality value for any node in a graph (based on binary or weighted measure)
g = the total number of actors in a network
a = the centrality of the central point in a star network of size *g*
b = the centrality of all non-central points in a star network of size *g*

Like the calculations for degree centralization, the denominator in this equation defines the maximum possible centralization score given the number of nodes in a network based on a star network. This measure ranges from 0 to 1 and represents the proportion of this maximum possible eigenvector centralization represented by a set of nodes. Our comments above concerning the achievability of a star network apply here also.

R Code for Calculating Network Measures

Calculation of Brainerd-Robinson similarity coefficients:

Input: frequency table sites (or other units of analysis) as rows and type categories as columns.

Example format:

	Type 1	Type 2	Type 3
Site 1	551	331	25
Site 2	10	300	62
Site 3	60	112	0
Site 4	12	10	14

```
sim.calc <- function(x) { # define function to calculate similarity scores
x <- prop.table(as.matrix(x),1)*100 # convert frequency table into row % matrix
rd <- dim(x)[1] # determine number of rows (sites) in table
results <- matrix(0,rd,rd) # create matrix with rows and columns for each site
for (s1 in 1:rd) { # open loop for s1 in 1 to the number of rows
for (s2 in 1:rd) { # open loop for s2 in 1 to the number of rows
x1Temp <- as.numeric(x[s1, ]) # lookup values in matrix x at row s1
x2Temp <- as.numeric(x[s2, ]) # lookup values in matrix x at row s2
results[s1,s2] <- 200 - (sum(abs(x1Temp - x2Temp)))}} # similarity of row s1 to s2
row.names(results) <- row.names(x) # add site names to row labels
colnames(results) <- row.names(x) # add site names to column labels
results <- results/200 # rescale similarity to range from 0 to 1
return(results)} # return the result of this function

sim.mat <- sim.calc(mydata) # run sim.calc function on frequency table (mydata)
```

Example `sim.mat` from `sim.calc` function using sample table above:

	Site 1	Site 2	Site 3	Site 4
Site 1	1	0.42	0.71	0.64
Site 2	0.42	1	0.68	0.47
Site 3	0.71	0.68	1	0.61
Site 4	0.64	0.47	0.61	1

Creation of Binary Network:

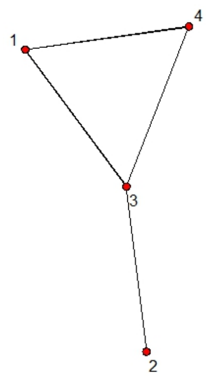
Requires R packages `statnet` and `sna`

To install package, type the following at the R console. This also installs the `sna` package.

```
install.packages('statnet')
```

Input: similarity matrix produced in the previous step – `sim.mat`

```
library(statnet)# initialize statnet library
cutoff <- 0.5 # set threshold for creating a tie to 0.5 similarity
bin <- event2dichot(sim.mat,method='absolute',thresh=cutoff) # binarize sim.mat at cutoff
rownames(bin) <- rownames(sim.mat) # apply row names from sim.mat to bin
colnames(bin) <- colnames(sim.mat) # apply column names from sim.mat to bin
net <- network(bin,directed=F) # create undirected network from binarized matrix bin
plot(net,displaylabels=T) # show network graph with sites labeled
```



The procedure above creates and displays a binary network from a similarity matrix (See above). To select a different binarization threshold, simply change the value for the `cutoff` variable.

Calculation of Centrality Scores for Binary Networks:

Input: R network produced in the previous step - `net`

```
net.stats <- function(y){ # define function to calculate centrality scores
dg <- as.matrix(degree(y,gmode='graph')) # calculate degree centrality
eg <- as.matrix(evcent(y)) # calculate eigenvector centrality
eg <- sqrt((eg^2)*length(eg)) # scale so sum of squared scores = number of nodes
bw <- betweenness(y,gmode='graph') # calculate betweenness centrality
output <- cbind(dg,eg,bw) # combine centrality scores into matrix
rownames(output) <- rownames(as.matrix(y)) # add row names to matrix
colnames(output) <- c('dg','eg','bw') # add column names to matrix
return (output)} # return results of this function

cent.scores <- net.stats(net) # run centrality scores for network (net)
```

Example `cent.scores` from `net.stats` function using sample table above:

	dg	eg	bw
Site 1	2	1.05	0
Site 2	1	0.56	0
Site 3	3	1.22	2
Site 4	2	1.05	0

Calculation of Centrality Scores for Weighted Networks:

Requires R packages `tnet and igraph`

To install packages, type the following at the R console (will install both packages).

```
install.packages('tnet')
```

Input: similarity matrix produced above – `sim.mat`

```
net.stats.wt <- function(y){ # define function to calculate weighted centrality scores
dg.wt <- as.matrix(rowSums(y)-1) # calculate weighted degree centrality
eg.wt <- as.matrix(evcent(y)) # calculate weighted eigenvector centrality
eg.wt <- sqrt((eg.wt^2)*length(eg.wt)) # scale so sum of squared scores = number of nodes
output <- cbind(dg.wt,eg.wt) # combine dg.wt and eg.wt centrality scores into matrix
rownames(output) <- rownames(as.matrix(y)) # add row names to matrix
colnames(output) <- c('dg','eg') # add column names to matrix
return (output)} # return results of this function

temp.scores <- net.stats.wt(sim.mat) # run centrality scores for similarity matrix

library(tnet) # initialize tnet library

sim.mat.t <- as.tnet(sim.mat) # convert similarity matrix to tnet object
bw.wt <- betweenness_w(sim.mat,directed=F,alpha=1) # calculate weighted betweenness
bw.wt <- as.matrix(bw.wt[,2]) # remove node labels
colnames(bw.wt) <- c('bw.wt') # add column name
cent.scores.wt <- cbind(temp.scores,bw.wt) # create matrix of weighted centrality scores

detach(package:tnet, unload=TRUE) # unload tnet package (required for other analyses)
```

Example `cent.scores.wt` from `net.stats.wt` function using sample table above:

	dg.wt	eg.wt	bw.wt
Site 1	1.77	1.01	0
Site 2	1.57	0.91	0
Site 3	2.00	1.09	0
Site 4	1.72	0.98	0

Calculation of network centralization:

Requires R packages `statnet`

```
library(statnet) # initialize statnet library

## calculate centralization measures for binary network (net) created above
centralization(net,degree,normalize=T) #calculate binary degree centralization
centralization(net,betweenness,normalize=T) #calculate binary betweenness centralization
centralization(net,evcent,normalize=T) #calculate binary eigenvector centralization

## calculate centralization measures for weighted network (sim.mat) created above
centralization(sim.mat,degree,normalize=T) #calculate weighted degree centralization
centralization(sim.mat,evcent,normalize=T) #calculate weighted eigenvector centralization
```

```
bw.cent <- function(x){ # define function for calculating
Cstar <- max(x) # determine maximum weighted betweenness centrality score in x
Csum <- (Cstar-x) # create vector or maximum centrality score minus each individual score
num <- 2*(sum(Csum)) # calculate numerator of centralization equation
den <- ((length(x)-1)^2)*(length(x)-2) # calculate denominator of centralization equation
out <- num/den # calculate centralization score
return(out)} # output result of this function

# calculate betweenness centralization for bw.wt variable in cent.scores.wt (column 3)
bw.cent(cent.scores.wt[,3])
```