Help File for the Maoz Network Program[1]

The Maoz Network Program is a Social Networks Analysis (SNA) package that implements many existing SNA programs, as well as new applications developed by Zeev Maoz. The programming was done with the help of Andrey Goder. This is a beta version that may well have numerous bugs. Users are encouraged to send questions, comments, and suggestions to zmaoz@ucdavis.edu.

In order to run the Network Program you will need the .NET 2005 compiler from Microsoft. You can download it from http://www.microsoft.com/downloads/details.aspx?FamilyID=9655156b-356b-4a2c-857c-e62f50ae9a55&DisplayLang=en.

This help file is divided into modules. Each module should be inserted as a separate help feature in the relevant menu items in the Network Program. The modules are organized by menu item.

FILE MENU

1. LOAD MENU

This program enables the user to load files in various formats. All files must be comma-delimited (.csv) files, however. The following formats can be loaded.

a. **Dyadic File format**: dyadic files consist of four columns as follows:

- Network identifier: four-digit integer index of the network. Each network has a different identifier.
- Row index: this index can be a numeric (integer) identifier, or an alphanumeric (string) identifier.
- Column index: same as row index. Column and row indexes must match exactly.
- Cell entry. Cell entry can be any real, integer, or binary number.
- **The user can specify if the first row is a labels row**

Example of a Dyadic File Format

Network Index	Row Index	Column Index	Cell Entry
2220	A	A	1
2220	A	B	1
2220	A	C	0
2220	A	D	1
2220	B	A	1
2220	B	B	1
2220	B	C	0
2220	B	D	1

[1] I would like to thank my research assistant Andrey Goder for helping programs the software package.

2220	C	A	0
2220	C	B	0
2220	C	C	1
2220	C	D	0
2220	D	A	1

b. **Matrix format**

A matrix file is a .csv file with the following structure.

- Top left cell, first matrix identifier
- Following matrix identifier, a *(n+1)* × *(n+1)* matrix. First row in the matrix provides column labels; first column of each matrix provides row labels. Entries in each matrix can be binary, integers, or real numbers denoting relations between row *i* and column *j*.
- Next matrix follows the same format starting with a network identifier on the top leftmost cell.

Note that the size of the matrix can change from one matrix to another.

Example of a matrix format:

1			**Column Index**			
Row Index	**1**	**2**	**3**	**4**	**5**	
1	0	0	1	1	1	
2	1	0	0	0	1	
3	0	1	0	0	0	
4	0	0	0	1	0	
5	1	0	1	1	1	
2			**Column Index**			
	1	**2**	**3**	**4**	**5**	**6**
1	0	1	0	0	1	0
2	1	0	1	1	0	1
3	1	1	1	1	1	0
4	1	1	1	1	1	0
5	0	1	0	0	0	0
6	0	0	0	1	1	1
3			**Column Index**			
	1	**2**	**3**	**4**	**5**	
1	0	0	0	1	0	
2	0	0	1	0	0	
3	0	0	0	0	0	
4	1	1	1	0	1	
5	1	1	1	1	0	
4			**Column Index**			
	1	**2**	**3**	**4**	**5**	
1	0	1	1	0	1	
2	1	0	1	1	1	
3	0	0	0	1	1	
4	0	0	0	0	0	
5	0	0	1	0	0	

Note: Example colors special cells of the matrices. In practice, cells are not colored or boldfaced. In this example green cells are network index; blue cells are row/column labels.

i. **Multi Variable Dyadic File**

Same format as single variable dyadic file except that it contains multiple variables in the columns following row and column labels.

Example

Network Index	Row Index	Column Index	Var1	Var2	Var3
2220	A	A	1	0.1	-3
2220	A	B	1	0.2	1
2220	A	C	0	0.1	-1
2220	A	D	1	0.3	0
2220	B	A	1	0.2	1
2220	B	B	1	0.1	-2
2220	B	C	0	0.3	-1
2220	B	D	1	0.4	2
2220	C	A	0	0.01	-3
2220	C	B	0	0.25	2
2220	C	C	1	0.4	-3
2220	C	D	0	0	-1
2220	D	A	1	0.1	0
2220	D	B	1	0.4	0
2220	D	C	0	0.7	1
2220	D	D	1	0.3	0

ii. **Multiple Matrix Format**

Same as single matrix format, with multiple matrices. This option requires a set of files each containing a group of sociomatrices with network identifiers (ni=1,…,k). The program requires the user to specify individual files where each file must have an identical range of matrix identifiers, and, for each matrix identifier, all matrices must be of the same dimension. For example, consider file *A* with matrices identified by 1, 2, 3, 5, 10, and with M_1 of size 3 × 3, M_{A2} of size 5 × 5, M_{A3} of size 7 × 7, M_{A5} of size 4 × 4, M_{A5} of size 6 × 6, and M_{A10} of size 3 × 3. The multiple matrix load program requires that files B, C, D…, have also the same matrix identifiers (1, 2, 3, 5, 10) and that each File M_{Bi}, M_{Ci}, M_{Di} have the same size as M_{Ai} (where i is the corresponding matrix identifier).

iii. **Affiliation Matrix format**

An affiliation matrix is an $n \times k$ matrix that connects nodes with events. Examples include lists of department members and their affiliation in professional societies, state membership in international organizations, religious or linguistic breakdowns of a state's population, and so forth. The format of an affiliation matrix is:

- Network identifier
- Row identifier
- Event1, event2, …, event *k*

An example of an affiliation network format is given by:

Network Identifier	Row Index	R1	R2	R3	R4	R5	R6	R7	R8	R9	
1	1	0	0	0	0	1	0	0	1	1	
1	2	1	1	0	0	1	0	0	0	0	
1	3	1	0	0	0	0	0	1	0	0	
1	4	0	0	1	1	0	1	0	1	1	
1	5	1	1	0	0	1	0	0	0	0	
1	6	1	1	1	0	1	1	0	0	1	
1	7	1	1	1	1	0	1	1	0	0	
1	8	0	1	0	1	0	1	0	0	0	
1	9	1	0	1	0	0	0	0	0	0	
1	10	0	0	0	1	0	0	1	1	0	
2	1	0	0	0	0	1	1	0	1		
2	2	1	1	1	1	1	1	1	1		
2	3	1	0	0	1	1	1	1	0		
2	4	1	1	0	0	1	1	0	1		
2	5	1	1	1	1	0	0	0	0		
2	6	0	0	1	1	0	0	1	0		
2	7	1	1	1	1	0	0	0	0		
2	8	1	0	0	0	1	0	0	0		
2	9	0	1	1	0	0	0	0	1		
2	10	0	0	0	0	0	1	1	0		
3	1	1	1	0	0	0	1	0	0	1	1
3	2	0	0	0	0	1	1	1	0	0	0
3	3	1	0	0	1	1	1	1	1	0	1
3	4	0	0	0	0	0	0	1	1	0	0
3	5	1	1	1	1	1	1	0	0	0	0
3	6	0	1	1	0	0	0	0	0	1	0
3	7	1	1	1	1	0	0	0	1	0	0
3	8	0	0	1	1	1	0	1	1	0	0
3	9	1	1	1	1	1	1	0	0	0	0
3	10	0	0	0	0	0	0	0	0	1	1

Note: Each network may have a different number of events. The first row in the dataset contains labels only for the first network. The network programs will assign labels to all networks regardless of the number of events.

iv. **Monadic Diagonal Files**

These are what SNA calls attribute files. The file forms an $n \times 1$ vector with the following format:

- Network identifier
- Row label
- Variable

An example is given by

Network Index	Row Index	Var1
2220	A	1
2220	B	1
2220	C	0
2220	D	1
2221	A	1
2221	B	1
2221	C	0
2221	D	1

The program converts the file into a diagonal sociomatrix with $s_{ii}=r_i$ and $s_{ij}=0 \ \forall j \neq i$.

v. **Random Matrices**

Random matrices are binary matrices that the program produces randomly from a range of $s_{ij}=0 \ \forall i,j \in N$ to $s_{ij}=1 \ \forall i,j \in N$ (where N is the size of the matrix's dimension). The user is asked to specify N and the beginning network index (defaults are $N=3$ and network index=1820). This input mode is suitable for simulations.

2. **SAVE AS MENU**

a. **Dyadic File.** File is saved as a dyadic file. The user is asked to specify the start and end values of the network identifier. The output is a .csv file with the following format:

- Network identifier
- Row label
- Column Label
- Value

An example output is:

Network Index	Row Index	Column Index	Cell Entry
2220	A	A	1
2220	A	B	1
2220	A	C	0
2220	A	D	1

2220	B	A	1
2220	B	B	1
2220	B	C	0
2220	B	D	1
2220	C	A	0
2220	C	B	0
2220	C	C	1
2220	C	D	0
2220	D	A	1
2220	D	B	1
2220	D	C	0
2220	D	D	1

b. **Matrix File:**

The program asks the user to specify the range of matrix identifiers to be written. The output file is a .csv file with the following format

- Matrix identifier
- Top row is column labels
- Rightmost column is row labels
- All other 2…n, rows and 2…n columns are matrix entries.

An example is:

1111			Column Index			
Row Index	**1**	**2**	**3**	**4**	**5**	
1	0	0	1	1	1	
2	1	0	0	0	1	
3	0	1	0	0	0	
4	0	0	0	1	0	
5	1	0	1	1	1	
1112			**Column Index**			
	1	**2**	**3**	**4**	**5**	**6**
1	0	1	0	0	1	0
2	1	0	1	1	0	1
3	1	1	1	1	1	0
4	1	1	1	1	1	0
5	0	1	0	0	0	0
6	0	0	0	1	1	1
1113			**Column Index**			
	1	**2**	**3**	**4**	**5**	
1	0	0	0	1	0	
2	0	0	1	0	0	
3	0	0	0	0	0	
4	1	1	1	0	1	
5	1	1	1	1	0	
1114						

	1	2	3	4	5
1	0	1	1	0	1
2	1	0	1	1	1
3	0	0	0	1	1
4	0	0	0	0	0
5	0	0	1	0	0

Note that the size of the matrix can change from one matrix to another.

c. **Network Characteristics Output File**. This is an output file of the counter program that provides general network attributes for each network. The format of the .csv output file is:

- **Network identifier**
- **N:** Network size
- **No. Clqs:** Number of cliques
- **Clq. Size:** Average Clique Size—Average number of members in cliques
- **Clq. Mem:** Average number of clique memberships per unit in networks
- **NPOL**: Node polarization index (defined in the matrix program)
- **CMOI**: Clique membership overlap index (defined in the matrix program)
- **Simple NPI**: Simple Network Polarization Index (defined in the matrix program)
- **COI**: Clique overlap Index (defined in the matrix program)
- **COIM:** Modified clique overlap index (defined in the matrix program).
- **NPOL Cohesion: NPOL** modified by average clique cohesion (defined in the matrix program)
- **NPI1 Cohesion: NPI** based on NPOL Cohesion and CMOI
- **NPI2 Cohesion: NPI** based on NPOL Cohesion and COI
- **NPOL Size: NPOL** based on clique size (specify external file in options menu).
- **NPOL Coh. Size: NPOL** based on both cohesion and clique size (external file)
- **NPI1 Coh. Size: NPI** based on NPOL Coh. Size and CMOI
- **NPI2 Coh. Size: NPI** based on NPOL Coh. Size and COI
- **Density**: Network density (defined in the matrix program)
- **Transitivity**: Transitivity index (defined in the matrix program)
- **Interdependence**: Systemic interdependence measure
- **Std**. **Interdependence**: Standardized systemic interdependence measure

d. **Multiple Matrix Files**. When a network procedure produces multiple matrix outputs, you can save it in multiple matrix files. The program tells you how many matrix files need to be written and prompts you to write the names of the files (as .csv files) successively. Once you finished inserting the names of the output files, it asks you to specify the network identifier range for the files.

e. **Multi Variable File**. When a network procedure produces multiple matrix outputs, you can save it in multiple variable dyadic file. The file has the same structure as the output single-variable file, except that the table has multiple variables. Example of a multivariable file is given below.

Network Index	Row Index	Column Index	Variable 1	Variable 2
2220	A	A	.3	1
2220	A	B	.5	1
2220	A	C	.7	0
2220	A	D	.1	1
2220	B	A	0	1
2220	B	B	0	1
2220	B	C	0	0
2220	B	D	.5	1
2220	C	A	.4	0
2220	C	B	0	0
2220	C	C	.5	1
2220	C	D	.4	0
2220	D	A	0	1
2220	D	B	0	1
2220	D	C	0	0
2220	D	D	0	1

3. **RESET Command**. This command empties the memory of the program, and requires it to reload input files. This command is useful after a program crash due to an error. Use it with caution.

4. **SCROLL MENU**

The program allows you to see the data in matrix/spreadsheet format. It presents each matrix on the screen separately. The scroll menu allows you to move from one matrix to another. It contains three options for moving:

a. **Next** (also goes with Ctrl+n key). Moves you to the next matrix in the MI sequence.
b. **Previous** (also goes with Ctrl+p key). Moves you to the previous matrix in the MI sequence.
c. **Jump** (also goes with the Ctrl+j key). Allows you to jump to a specified MI sequence.

5. **OPTIONS MENU**

This menu defines the central options that are used to derive various SNA statistics. You may change the options at any time during the session. The change of options will affect subsequent procedures.

The Options menu consists of the following options:

a. **Cutoffs for clique extraction**. The default value for defining a relationship in the clique extraction program is 0.0. This means that any relationship that has a value of $s_{ij}>0$ is treated as 1. You can change this value manually, by inserting a number different from zero in the appropriate window. In this case, the program will use the same cutoff value for all the matrices in the dataset. Alternatively, if you wish to use different cutoffs for different networks in the dataset, you can define an external (.csv) file that has the following structure:

Network	Cutoff
2220	0.2
2221	0.5
2222	0.1
2223	0.8

You insert the file name at the prompt for Use Cutoff file.

b. **Max matrix density value**. This defines the maximum value for calculation of density scores. Here too, you can specify a value in the window next to the option; this value will apply to all matrices in the dataset. Or you can use an external (.csv) file that has the following structure:

Network	Max density value
2220	1
2221	0.7
2222	55
2223	0.3

This will define varying maximum density values for each matrix according to its network identifier.

c. **Cohesion Matrix Source**. This option allows you to specify whether to use clique cohesion indices in the calculations of **NPOL** (see matrix menu), and if so—to define the source from which cohesion data will be obtained. There are several options here:

i. None (do not use cohesion scores)

ii. External matrix file. The file should have the same format as the input matrix format and its (Network Identifiers) NIs should match exactly those of the NIs in the current input file. If you choose this option, you must click the **Select File** command next to the option and specify the matrix file containing the cohesion scores.

iii. External dyadic file. The file should have the same format as the input dyadic file and its NIs should match exactly those of the NIs in the current input file. If you choose this option, you must click the **Select File** command next to the option and specify the dyadic file containing the cohesion scores.

iv. Structural equivalence. This option uses the current input file as the source for generating standardized Euclidean distance structural equivalence scores. You do not need to specify an external file here.

v. Use **NPOL Size**. In order to calculate **NPOL** based on an attribute file that specifies the relative size of units in each clique—rather than the number of units in each clique—you need to select an external (.csv) file that has a monadic format from which relative sizes of units are derived. The monadic file should have the following format

Network Index	Row Index	Unit Rel. Size
2220	A	0.5
2220	B	0.3
2220	C	0.1
2220	D	0.1
2221	A	0.1
2221	B	0.1
2221	C	0.4
2221	D	0.4

vi. Use **NPOL Coh. Size.** This option specifies the use of both cohesion scores (that may be obtained from an external file or defined endogenously—see above in the Cohesion matrix source) and size data derived via a user selected relative size file, as for **NPOL Size**.

d. **Sum/Mean Attributes**. When you select files that denote the attributes of the cliques, the program defines the clique attributes on the basis of user specified statistics. The external vector or matrix files contain characteristics of units or of dyads. The program aggregates these characteristics into a clique-level characteristic. The nature of this aggregation may be specified by the user. Several options are available.

  i. None. The default option.

  ii. Mean. This generates the mean of the Clique Characteristics Vector (CCV) or the mean of all dyadic values in a given clique if you specify a Dyadic Clique Characteristics Matrix (DCM).

  iii. Sum. This generates the sum of the Clique Characteristics Vector (CCV) or the sum of all dyadic values in a given clique if you specify a Dyadic Clique Characteristics Matrix (DCM).

e. **Transitivity Type.** There are variable options for generating transitivity scores. The typical SNA transitivity conception is the **simple link** conception. A triad is considered transitive ($t_{ijk} = 1$) if $s_{ij}>0$ and $s_{ik}>0$ and $s_{jk}>0$. This is also the simple link option here. There are, however, two other options. These are:

i. **Weak Link Transitivity.** t_{ijk} = 1 if $s_{ij}>0$, $s_{ik}>0$ and $s_{jk}>min(s_{i(jk)})$ where $min(s_{i(jk)})$ is the minimum value of the link between *i* and *j* or *i* and *k*.

ii. **Strong Link Transitivity.** t_{ijk} = 1 if $s_{ij}>0$, $s_{ik}>0$ and $s_{jk}>max(s_{i(jk)})$ where $max(s_{i(jk)})$ is the maximum value of the link between *i* and *j* or *i* and *k*.

f. Clique Extraction Options. Given a nonsymmetrical matrix ($s_{ij} \neq s_{ji}$), the clique extraction algorithm requires symmetricizing the matrix. This set of options defines the rules for symmetrization.

i. Option 1 **max.** The Semmetricized matrix S' is defined such that for each cell of S' (s'_{ij}) we have:

$$s'_{ij} = s'_{ij} \begin{cases} 1 & if\ s_{ij} > c\ or\ s_{ji} > c \\ 0 & otherwise \end{cases}$$

where c is the cutoff value specified above. This implies that whenever one of the cells in S, s_{ij} or s_{ji} assumes a value exceeding the cutoff, the symmetricized matrix assigns a value of 1 to both corresponding cells.

ii. Option 2 **upper.** The symmetricized matrix *S'* is defined such that for each cell of *S'* (s'_{ij}) we have:

$$s'_{ij} = s'_{ji} = \begin{cases} 1 & if\ s_{ij|j \geq i} > c \\ 0 & otherwise \end{cases}$$

where c is the cutoff value specified above and $i, j \in N$ are index numbers of rows and columns in the S matrix respectively. This implies that, regardless of the value of s_{ji}, the value of both s'_{ij} and of s'_{ji} assume the value of *1* or 0 based on whether s_{ij} exceeds the cutoff, but this applies only to cells whose column index number is equal to or higher than the row index number.

iii. Option 3 **lower.** The symmetricized matrix *S'* is defined such that for each cell of *S'* (s'_{ij}) we have:

$$s'_{ij} = s'_{ji} = \begin{cases} 1 & if\ s_{ij|j \leq i} > c \\ 0 & otherwise \end{cases}$$

where c is the cutoff value specified above and $i, j \in N$ are index numbers of rows and columns in the S matrix respectively. This implies that, regardless of the value of s_{ji}, the value of both s'_{ij} and of s'_{ji} assume the value of *1* or 0 based on whether s_{ij} exceeds the cutoff, but this applies only to cells whose column index number is equal to or lower than the row index number.

iv. Option 4 **min.** The Semmetricized matrix S' is defined such that for each cell of S' (s'_{ij}) we have:

$$s'_{ij} = s'_{ij} \begin{cases} 1 & if\ s_{ij} > c\ \&\ s_{ji} > c \\ 0 & otherwise \end{cases}$$

where c is the cutoff value specified above. This implies that only when both the cells in S, s_{ij} and s_{ji} assumes a value exceeding the cutoff, the symmetricized matrix assigns a value of 1 to both corresponding cells.

g. Networks Characteristics Output. This set of options specifies which of the network characteristics you wish to calculate in the Network Characteristics program in the matrix menu. The options are:

- **Network identifier**
- **N:** Network size (number of units in the network)
- **No. Clqs:** Number of cliques
- **Clq. Size:** Average Clique Size—Average number of members in cliques
- **Clq. Mem:** Average number of clique memberships per unit in networks
- **NPOL**: Node polarization index (defined in the matrix program)
- **CMOI**: Clique membership overlap index (defined in the matrix program)
- **Simple NPI**: Simple Network Polarization Index (defined in the matrix program)
- **COI**: Clique overlap Index (defined in the matrix program)
- **COIM:** Modified clique overlap index (defined in the matrix program).
- **NPOL Cohesion: NPOL** modified by average clique cohesion (defined in the matrix program)
- **NPI1 Cohesion: NPI** based on NPOL Cohesion and CMOI
- **NPI2 Cohesion: NPI** based on NPOL Cohesion and COI
- **NPOL Size: NPOL** based on clique size (specify external file in options menu).
- **NPOL Coh. Size: NPOL** based on both cohesion and clique size (external file)
- **NPI1 Coh. Size: NPI** based on NPOL Coh. Size and CMOI
- **NPI2 Coh. Size: NPI** based on NPOL Coh. Size and COI
- **Density**: Network density (defined in the matrix program)
- **Transitivity**: Transitivity index (defined in the matrix program)
- **Interdependence**: Systemic interdependence measure
- **Std**. **Interdependence**: Standardized systemic interdependence measure

4. **Standardize Menu**

The input data can be standardized into one of the following types:

a. **None.** Default. Data are unstandardized.

b. **Row Standardization**. Each entry s_{ij} in the matrix is standardized such that $s'_{ij} = s_{ij} / s_{.i} = s_{ij} \Big/ \sum_{j=1}^{n} s_{ij}$. Hence each row-standardized cell entry is the unstandardized entry's proportion of its respective row.

c. **Column Standardization**. Each entry s_{ij} in the matrix is standardized such that $s'_{ij} = s_{ij} / s_{i.} = s_{ij} \Big/ \sum_{i=1}^{n} s_{ji}$. Hence each row-standardized cell entry is the unstandardized entry's proportion of its respective column.

d. **Diagonal Standardization.** Each entry is standardized such that the entry is divided by its respective column diagonal. Hence, $s'_{ij} = s_{ij} / s_{jj}$.

6. **MATRIX MENU**

The Matrix Menu performs most of the SNA operations. It is useful before using this menu to go to the options menu and specify whatever options you need in order to perform various operations. **Note.** Operations in this menu are performed on the current sociomatrix or affiliation matrix that is on display. To move to another matrix use the **Scroll** menu. When scrolling, the current operation will be automatically performed on the matrix to which you have chosen to move.

a. **Default.** Data are presented in the input format. If you choose to standardize the data, they will be standardized according to the standardize option selected, and subsequent matrix operations will be performed on the standardized data.

b. **Clique Affiliation (CA) Matrix.** This procedure extracts cliques according to the **clique extraction options** specified in the **Options** menu. The output in the matrix window is the $n \times k$ clique affiliation (CA) matrix of the current matrix in the data file. Each entry ca_{ij} is 1 if row-unit i is a member of column-clique j, and zero otherwise. You can use the **scroll menu** to scroll through clique affiliation matrices, and use the **save as** menu to save the clique affiliation matrices.

c. **Clique Member Overlap (CMO) Matrix.** This is a $n \times n$ matrix (obtained as $CMO = CA \times CA'$) that has the following characteristics: (i) $cmo_{ij}=cmo_{ji}$ denotes the number of cliques that units i and j share in common, and (ii) cmo_{ii} is the number of cliques that have unit i as a member. You can use the **scroll menu** to scroll through clique member overlap matrices, and use the **save as** menu to save the clique member overlap matrices.

d. **Clique-by-Clique Overlap (CCO) Matrix.** For a network of size m with a clique affiliation (CA) matrix of dimensions $n \times k$, the clique-by-clique overlap (obtained as: $CCO=CA'\times CA$) is a matrix of dimensions $k \times k$ that has the following characteristics: (i) $cco_{ij}=cco_{ji}$ is the number of units that cliques i and j share in common, and (ii) cco_{ii} is the number of members in clique i. You can use the **scroll menu** to scroll through clique-by-clique overlap matrices, and use the **save as** menu to save the clique-by-clique overlap matrices.

e. **Dependency Matrix (D).** This matrix produces dependency data using the procedure described in Maoz (2006a and 2006b). For a sociomatrix S of order n, the dependency procedure requires the user to specify the number of iterations for the reachability matrix. This number can range from 1 (which uses the first-order sociomatrix $S=S^1$), 2 (which uses the sum of the first-order (S^1) and second-order (S^2) matrices, and up to $n-1$ ($D=S^1+S^2+\ldots S^{n-1}$). The resulting matrix is a $(n+3) \times (n+3)$ matrix with that has the following characteristics: (i) $d_{ij}\neq d_{ji}$ denotes the dependence of the column entry j on the row entry i, (ii) d_{ii} denotes the self-dependence of unit i, (iii) Entry $d_{i(n+1)}$ is the total outdependence of unit i, that is, the total dependence of all units in the network on unit i (including i's self-dependence), (iv) Entry $d_{i(n+2)}$ is the standardized outdependence of unit i, defined as: $stoutd_i = (\sum_{j=1}^{n} d_{ij} - d_{ii}) \Big/ \sum_{j=1}^{n} d_{ij}$, (v) Entry $d_{i(n+3)}$ is the *raw outdependence* defined as $rwoutd_i = \sum_{j=1}^{n} d_{ij} - d_{ii}$, and defines the total dependence of all units in the network on unit *i,* excluding *i'*s self-dependence. (vi) Entry $d_{(n+1)j}$ is the *total ondependence* of unit *j*, which means the total dependence of unit *j* on other units in the system, (vii) Entry $d_{(n+2)j}$ is the *standardized ondependence* calculated in the same way as the *standardized outdependence* for the columns of the matrix. (viii) Entry $d_{(n+3)j}$ is the *raw ondependence* defined in the same way as *rwoutd* for the appropriate column. (ix) Entry $d_{(n+2)(m+2)}$ is the system average of $sdtoutd_i$ and $stdond_j$ and (x) Entry $d_{(n+3)(n+3)}$ is the system average of $rwoutd_i$ and $rwond_j$. Dependency matrices can also be scrolled via the **scroll menu** or they can be saved via the **save as** menu.

f. **Reachability Matrix (R)**. The reachability matrix is an $n \times n$ matrix whose entries r_{ij} specify by how many paths unit *j* could be reached from unit *i.* Diagonal entries, r_{ii} specify reachability *cycles*, paths leading from a unit to itself through other units. This procedure allows the user to specify the number of iterations used for the reachability matrix. For example, given a sociomatrix S, specifying 3 in the option implies that $R = S^1 + S^2 + S^3 = \sum_{i=1}^{3} S^i$. Likewise, specifying 5 in the option implies that $R = \sum_{i=1}^{5} S^i$.

g. **Single Matrix Structural Equivalence**. This program produces structural equivalence scores for single matrices. Given a sociomatrix *X*, there are three options for producing structural equivalence scores.

i. **Correlations.** The output of this procedure is a structural equivalence matrix of Pearson product-moment correlation coefficients, SE_c. Each entry in this matrix is defined as:

$$se_{ij} = se_{ji} = \frac{\sum_{k=1}^{n}(x_{ik} - \overline{x}_{\bullet i})(x_{jk} - \overline{x}_{\bullet j}) + \sum_{k=1}^{n}(x_{ki} - \overline{x}_{i\bullet})(x_{kj} - \overline{x}_{j\bullet})}{\sqrt{\sum_{k=1}^{n}(x_{ik} - \overline{x}_{\bullet i})^2 + \sum_{k=1}^{n}(x_{ki} - \overline{x}_{i\bullet})^2}\sqrt{\sum_{k=1}^{n}(x_{jk} - \overline{x}_{\bullet j})^2 + \sum_{k=1}^{n}(x_{kj} - \overline{x}_{j\bullet})^2}}$$

where $\overline{x}_{\bullet i}$ is the mean of the *i*th row of the matrix and $\overline{x}_{i\bullet}$ is the mean of the *i*th column of the matrix ($\overline{x}_{j\bullet}$ *and* $\overline{x}_{\bullet j}$ are defined in the same way for the *j*th row and column, respectively). The se_{ij} scores here are defined in the range [-1,+1] where higher *se* scores indicate *higher* structural equivalence.

ii. **Raw Euclidean Distance**. The output of this procedure is a structural equivalence matrix of raw Euclidean distances. Each entry in the output matrix SE_{reu} is the raw Euclidean distance score between the row and column, and is defined by:

$$se_{ij} = se_{ji} = \sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2 + \sum_{k=1}^{n}(x_{ki} - x_{kj})^2}$$

Note that higher se_{ij} scores in this option are defined in the range of [0, ∞] with higher *se* scores indicating *lower* level of structural equivalence.

iii. **Standardized Euclidean Distance.** The output of this procedure is a structural equivalence matrix of standardized (in the range of [0,1]) Euclidean distance scores. The entries here are defined as:

$$se_{ij} = se_{ji} = 1 - \frac{\sqrt{\sum_{k=1}^{n}(x_{ik} - x_{jk})^2 + \sum_{k=1}^{n}(x_{ki} - x_{kj})^2}}{n^2 \max(x_{ik} - x_{jk})}$$

Where $\max(x_{ik} - x_{jk})$ is the maximum distance between any two entries in the matrix. Here structural equivalence increases with the size of se_{ij}.

Note: output SE matrices can also be scrolled via the **scroll menu** or they can be saved via the **save as** menu.

h. **Multi-matrix Structural Equivalence**

This program is equivalent to the simple structural equivalence program, for multiple matrices. For example, consider a set of $\Re = [r_1, r_2, \ldots, r_r]$ sociomatrices of size $n \times n$, all having the same units in the same order. Suppose r_1 is a friendship matrix (with entry r_{1ij} indicating whether person *i* is a friend of person *j*; r_2 is a neighborhood matrix (entry r_{2ij} indicates whether person *i* lives in the same block as person *j*), and r_3 is a soccer team membership matrix (r_{3ij} indicates whether person *i* is in the same soccer team as person *j*). The program measures structural equivalence for any pair of units across all three matrices. The options here are the same as for the single-matrix structural equivalence program:

i. **Multiple Correlation Structural Equivalence.** This is given by:

$$SE_{ij}^{R} = \frac{\sum_{r=1}^{2R}\sum_{k=1}^{n}(x_{ikr} - \bar{x}_{ir\bullet})(x_{jkr} - \bar{x}_{jr\bullet})}{\sqrt{\sum_{r=1}^{2R}\sum_{k=1}^{n}(x_{ikr} - \bar{x}_{ir\bullet})^2}\sqrt{\sum_{r=1}^{2R}\sum_{k=1}^{n}(x_{jkr} - \bar{x}_{jr\bullet})^2}}$$

where $R=[1,\ldots,r]$ is a set of relationships. Since each matrix *r* can be transposed, the multiple correlation includes the transpose of the relations. The *SE* scores here are defined in the range [-1,+1] where higher *SE* scores indicate *higher* structural equivalence.

ii. **Euclidean Distance Structural Equivalence.** This is given by:

$$SE_{ij}^{R} = SE_{ji}^{R} = \sqrt{\sum_{r=}^{r}\sum_{k=1}^{n}[(x_{ikr} - x_{jkr})^2 + (x_{kir} - x_{kjr})^2]}$$

Note that higher se_{ij} scores in this option are defined in the range of [0, ∞] with higher *se* scores indicating *lower* level of structural equivalence.

iii. **Standardized Euclidean Distance Structural Equivalence.** This is given by

$$SE_{ij}^{R} = SE_{ji}^{R} = 1 - \frac{\sqrt{\sum_{r=1}^{R}\sum_{k=1}^{n}(x_{ikr} - x_{jkr})^2 + (x_{kir} - x_{kjr})^2}}{Rn^2\sum_{r=1}^{R}\max(x_{ikr} - x_{jkr})}$$

Here structural equivalence varies in the range of [0,1] and increases in SE_{ij}.

i. **Centrality Indices Matrix**. This program calculates various centrality scores of units of a given sociomatrix. Before calculating these indices, the program requires the user to define three options: (1) the value of $s_{ij}(max)$, that is the maximum value of any of the entries in the sociomatrix *S,* (2) in betweenness centrality (defined below) it requires the user to specify whether cycles are to be included, and (3) in the course of generating centrality indices, for some purposes diagonal values need to be zeroed (self-links are ignored); the option asks whether to do this in cases where sociomatrices have nonzero diagonal elements.[2] The centrality indices program contains the following measures of centrality.

i. **Degree Centrality:** Degree centrality is the centrality of a given node in terms of the number of nodes to which it is directly connected. There are two measures of centrality **OutDegree Centrality**, measured for each node $i \in N$ as: $DO_i = \sum_{j=1}^{n} s_{ij} \Big/ n[\max(s_{ij})]$. This measures the centrality of outgoing links between a node *i* and other nodes in the network. **InDegree Centrality (**that is also used as a measure of *Degree Prestige*)**,** measured for each node $j \in$ N as: $DO_j = \sum_{i=1}^{n} s_{ji} \Big/ n[\max(s_{ij})]$.

ii. **Closeness Centrality.** This measures centrality in terms of the closeness of nodes (the number of vertices separating them defines their closeness). Here too we have **OutCloseness**, defined as $CO_i = (\sum_{j=1}^{n} r_{ij} - r_{ii})/[(n-1)\max(s_{ij})]$ (where r_{ij} is an element of the reachability matrix *R, j* indexes columns, and $s_{ij(max)}$ is specified by the user), and **InCloseness**, defined as

2 A cycle is a link of a node to itself through at least one other node.

$CI_j = (\sum_{i=1}^{n} r_{ji} - r_{jj}) / [(n-1)\max(s_{ij})]$ (where r_{ij} is an element of the reachability matrix *R*, *i* indexes rows and $s_{ij(max)}$ is specified by the user).

iii. **Betweenness Centrality**. A node *i* is said to have a betwenness value of 1 if $s_{ij} \neq 0$ and $s_{ik} \neq 0$. In this case, *i bridges between j and k*. Betweenness centrality measures the ratio of cases where a given node serves as a bridge between two other nodes and the possible number of bridges for this node. Here too we have **Out Betwenness Centrality,** measuring only the links where the outgoing ties of a given node serve as a bridge between other nodes, and **InBetwenness Centrality** that measure incoming nodes.

iv. **Eigenvector Centrality**. This index weights the degree centrality scores of a node by the degree centrality of the nodes to which it is connected. Here too we have **OutEigenvector Centrality**, based on outgoing ties, and **InEigenvector Centrality**, based on the incoming ties.

The output for this program is a $n \times 8$ table that lists the **Out** and **In** centrality scores for each of the *n* nodes.

j. **Clique Characteristics Matrix**. This program allows weighing cliques by various characteristics as specified in the **Options→Sum/Mean Attributes**. The output is a $k \times (m+2)$ table where the *k* rows represent the cliques in the network, the leftmost column is the NI, the next column is the number of members in clique *i*, and the following column represent for each DVC or CCM file specified in the options for this program the attributes of the clique for the variable specified in this file.

k. **Affiliation to Matrix Conversion.** This procedure converts affiliation matrices to sociomatrices. There are three possible ways to do this:

i. **Sociomatrix conversion**. This is the traditional SNA approach to conversion. Given an affiliation matrix *A* of size $n \times k$, the conversion is done such that the resulting sociomatrix *S* (of size $n \times n$) is: $S = A \times A'$. Each entry in *S*, s_{ij} is the number of events common to nodes *i* and *j*. Diagonal entries s_{ii} denote the number of events in which *i* is a member.

ii. **Correlation conversion**. This method converts an affiliation matrix *A* of size $n \times k$ into a sociomatrix *S*, whose entries s_{ij} vary in the [-1,+1] range, and reflect Pearson product-moment correlations between rows *i* and *j* in matrix *A* (diagonal entries s_{ii} are correlations of nodes with themselves, thus designated values of 1, by definition).

iii. **Standardized Euclidean distance**. This method converts an affiliation matrix *A* of size $n \times k$ into a sociomatrix *S*, whose entries s_{ij} vary in the [0,1] range and reflect standardized Euclidean distances between rows *i* and *j* in matrix *A* such that,

$$s_{ij} = s_{ji} = 1 - \frac{\sqrt{\sum_{k=1}^{n}(a_{ik} - a_{jk})^2}}{n[\max(a_{ik} - a_{jk})]}$$

(diagonal entries s_{ii} are distances of nodes with themselves, thus designated values of 1, by definition).

l. **Event Overlap Matrix**. This procedure converts an affiliation matrix A of order $n \times k$ into an event overlap matrix E of order $k \times k$, defined as $E = A' \times A$. Entries in E, e_{ij} reflect the number of members that cliques i and j share in common. Diagonal entries e_{ii} reflect the number of members in clique i.

m. **Unit Dependency Matrix**. This procedure produces unit dependency data (see **Dependency Matrix**) in an $n \times 6$ output matrix. The output matrix is organized as follows.

Network Identifier	Unit	*Rwoutd*	*Rwind*	*Stdoutd*	*Stdind*

n. **Network Characteristics Data**. This procedures produces the **network characteristics data** for the current sociomatrix that are specified in the **options** menu.

o. **Matrix Multiplication**. This is a utility program allowing simple matrix multiplication of a series of matrices. It uses the current dataset that is composed of one or more matrices (of a general $m \times n$ order, with m and n varying from one input matrix to another), and asks the user to specify another matrix (that can be inputted either as a matrices dataset or as a dyadic dataset). It then performs the multiplication operation.

Example. Suppose the input dataset consists of the following matrices, $S_{1(3\times2)}$, $S_{2(4\times5)}$, $S_{3(7\times3)}$ (expressions in the subscripted parentheses denote the dimensions of each matrix). In order to multiply this dataset by another set of matrices O, the dataset to be inputted in the option must be: $O_{1(2\times a)}$, $O_{2(5\times b)}$, $O_{3(3\times c)}$ (where a, b, and c are any positive integers). Otherwise the procedure will crash. The output screen presents the first product matrix $T_{1(3\times a)} = S_1 \times O_1$. Other matrices in the resulting dataset can be displayed via the **Scroll Menu**.

p. **Elementwise Multiplication**. This procedure performs elementwise multiplication of the current matrix S by a matrix M or by a vector V that are specified by the user. The result is an elementwise multiplication of s_{ij} by m_{ij} or by v_i. For the procedure to work, the matrix M must be of the same dimensions as matrix *S*, or the vector must have the same number of rows as does *S*. The user can specify three forms of input for the multiplied datasets: (a) matrix file, (b) dyadic file, or (c) monadic file (vector).

Example. Given a dataset consisting of the following matrices $S_{1(3\times2)}$, $S_{2(4\times5)}$, $S_{3(7\times3)}$ (expressions in the subscripted parentheses denote the dimensions of each matrix), the dataset used for elementwise matrix multiplication must be $O_{1(3\times2)}$, $O_{2(4\times5)}$, $O_{3(7\times3)}$. If the user chooses a vector option, then the vectors must be $V_{1(3\times1)}$, $V_{2(4\times1)}$, $V_{2(7\times1)}$. The output screen presents the first product matrix $T_{1(3\times a)} = S_1 \times O_1$. Other matrices in the resulting dataset can be displayed via the **Scroll Menu.**

q. **Binary Complement**. This procedure produces a binary complement of a sociomatrix *S* of order $n \times m$ a binary matrix that is a complement of the binary version of the *S* matrix. Let *SB* is the binary version of the matrix *S* as defined by the cutoff procedure in the **Options** menu. The binary complement matrix *BC* is a matrix such that $SB–BC=0$.

r. **Inter-Clique Distance Matrix**. This procedure examines the relative distance between any two cliques in a given network in terms of the pattern of membership between a given clique c_i and all other cliques $c_{j\neq i}$. This is in fact a standardized Euclidean distance matrix performed on the **CCO** matrix that is derived from the sociomatrix S.

s. **CONCOR Matrix**. Given a sociomatrix *S* the Convergence of Correlations (CONCOR) procedure generates blocks (mutually exclusive subsets) of all units in the network based on iterative convergence of structural equivalence scores based on correlations. It proceeds in the following steps: (1) The user specifies a correlation cutoff that serves as the minimum correlation between units *i* and *j* (se_{ij}) that assigns these units to the same block, (2) the program generates the *SE* correlation matrix and assigns all units that have correlations

t. **Triadic Data Matrix**. From any sociomatrix *S*, this procedure lists all possible nondirectional triads in the matrix with all dyadic and triadic relational patterns that characterize each of the triads. This characterization is based on a table derived from Burt (1990: 87). Each triad can be characterized by one of 36 types of ties (that include positive and/or negative signs). For example, consider the following matrix.

	1	2	3	4	5
1	0	0	1	0	0
2	0	1	0	0	0
3	0	0	1	0	0
4	0	1	1	1	1
5	0	0	0	1	1

The output of the triadic data matrix procedure takes all triads in this matrix, and defines: first, the relationship between all of the dyads making up a given triad, and then defines the "Burt" type of the triad by assigning it the type identifier in Burt's list. The output of the procedure for the matrix shown above is given below.

MI	Triad	A→B	A→C	B→A	B→C	C→A	C→B	ABC	ACB	BAC	BCA	CAB	CBA
1820	123	0	1	0	0	0	0	2	2	21	21	4	4
1820	124	0	0	0	0	0	1	21	21	4	4	2	2
1820	125	0	0	0	0	0	0	1	1	1	1	1	1
1820	134	1	0	0	0	0	1	31	22	5	5	22	31
1820	135	1	0	0	0	0	0	2	2	4	4	21	21
1820	145	0	0	0	1	0	1	11	11	6	6	6	6
1820	234	0	0	0	0	1	1	32	24	32	24	3	3
1820	235	0	0	0	0	0	0	1	1	1	1	1	1
1820	245	0	0	1	1	0	1	14	14	9	9	26	26
1820	345	0	0	1	1	0	1	14	14	9	9	26	26

Note that each triad has six permutations. The triadic columns list all six permutation in the order of elements in the triad for the appropriate row. Thus, for triad 245 the corresponding columns mean: ABC=245, ACB=254, BAC=425, BCA=452, CAB=524, CBA=542.

u. **Role Equivalence Matrix**. Following Burt (1990: 86) role equivalence measures the equivalence, defined as a comparison between any two units in the network in terms of the convergence of their pattern of roles in the network as "a triad census, a pattern of relative frequencies with which kinds of triads describe the individual's orientation to others." Two individuals are said to be "role equivalent" to the extent that the distribution of their triadic census positions with others is identical. The current procedure uses standardized Euclidean distances between the role positions of any two units in the networks.

Bibliography

Ronald S. Burt 1990. Detecting Role Equivalence. *Social Networks*, 12(1): 83-97.

Zeev Maoz 2006a. The Effects of Strategic and Economic Interdependence on International Conflict: A Cross-Levels-of-Analysis Study. Mimeographed. University of California, Davis.

——— 2006b. Systemic Polarization, Interdependence, and International Conflict, 1816-2002. *Journal of Peace Research*, 43(4): 391-411.