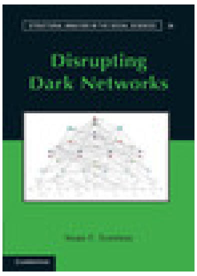


Disrupting Dark Networks

Sean F. Everton

Book DOI: http://dx.doi.org/10.1017/CBO9781139136877

Online ISBN: 9781139136877

Hardback ISBN: 9781107022591

Paperback ISBN: 9781107606685

Chapter

4 - Gathering, Recording, and Manipulating Social Networks pp. 76-132

Chapter DOI: http://dx.doi.org/10.1017/CBO9781139136877.007

Cambridge University Press

4

Gathering, Recording, and Manipulating Social Networks

4.1 Introduction

Social network analysis software packages such as UCINET, NetDraw, Pajek, and ORA often come with ready-to-use data. In the real world, however, we have to collect and record it on our own. Knowing how to do this is an important first step in SNA. We can gather and prepare social network data in a variety of ways, but most social network analysts record social network data in matrix form. For small datasets it usually makes sense to enter the data directly into UCINET or ORA, using their internal spreadsheet functions. When working with large networks, however, it is generally preferable to initially enter social network data into a standard spreadsheet program, such as Microsoft Excel, and then import (or paste) it into UCINET, NetDraw, Pajek, or ORA. One of Excel's attractions is its "auto-complete" feature that compares the text you are typing into a cell with text already entered into the same column, which increases accuracy (e.g., consistently spelling the same name the same way each time) and input time.[1]

This chapter focuses on collecting, recording, and manipulating network data. Before turning to the nuts and bolts of how to do this, however, we need to first consider how social network analysts specify a network's boundaries, the difference between personal (ego) and complete networks, the various types of social network data, and the variety of ways that researchers collect social network data. Only then will we be ready to collect, record, and manipulate actual social network data.

[1] If the same word has been used before, it completes typing the entry for you.

4.2 Boundary Specification

An important concern in social network study is which actors to include in the network and which ones not to include. Sometimes it is relatively clear. Sometimes it is not. The goal should be "to find a set of actors with relatively good separateness from the rest of the world" from which we can draw reasonable conclusions (Erickson 2001:317). Laumann, Marsden, and Prensky (1983, 1989) note that researchers adopt various approaches for determining the boundaries of their networks. Generally, their approaches cut across two dimensions: one that leads them to choose between realist or nominalist strategies, and another that causes them to focus on one of three aspects of a network such as actor attributes, relations, or participation in activities/events. We consider these in turn.

Realist vs. Nominalist Strategies

The *realist* approach is a more subjective strategy for identifying the boundaries of a network. It allows actors to define the boundary of the network. It seeks to adopt the "vantage point of the actors themselves ... [and] the network is treated as a social fact only in that it is consciously experienced ... by the actors composing it" (Laumann et al. 1983:20). It assumes that the network exists as a social entity for most (or perhaps all) actors of the network (Laumann, Marsden, and Presnsky 1983). Thus, actors and their ties are only included to the extent that other actors consider them to be part of the network (Knoke and Yang 2007:15). This approach implicitly assumes that natural boundaries actually exist for the network. Knoke and Laumann (1982) adopted this approach when identifying core U.S. energy and health national policy organizations for analysis; they only included those organizations that energy and health policy insiders considered to be influential players in setting energy and health policies in the United States (Knoke and Yang 2007:15–16).

> The key aspect of the realist approach is identifying, early on, who are the key informants [in the] target population. You need to be confident that the informants you speak with are ones who have a good understanding of the network in question and can offer an accurate picture of the members of the network. Thus, locating who is a "key informant" might be a tricky process in itself. (Prell 2011:66)

By contrast, the *nominalist* approach is a more objective strategy in that rather than looking to the perceptions of network members, it imposes an a priori framework based on the analyst's theoretical concerns

(Knoke and Yang 2007; Wasserman and Faust 1994). "For example, a researcher might be interested in studying the flow of computer messages among researchers in a scientific specialty. In such a study, the list of actors might be the collection of people who published papers on the topic in previous years. The list is constructed for the analytical purposes of the researcher, even though the scientists themselves might not perceive the list as constituting a distinctive social entity" (Wasserman and Faust 1994:32). In essence it imposes somewhat arbitrary boundaries on a network based on the needs of the researcher. For example, researchers might choose to only examine the social networks of second-grade children at a particular school; even though these second-graders have ties outside of that group, they are ignored for the purposes of the study. However, if we define a network's boundary in this way, we need to be aware that we could "be potentially ignoring important ties that influence the behaviours of each participant in [our] study . . . Thus [we] need to be able to fully justify [our] reasons for drawing a boundary . . . based on this approach" (Prell 2011:67).

Although it is analytically useful to draw a distinction between these two approaches, it is probably better to think of them as two poles of a continuum where it is possible to imagine a strategy that adopts a little of both approaches in defining a network's boundary. For example, you might begin with a nominalist approach in studying a dark network by drawing on court proceedings and newspaper accounts to initially define the network. But then you could supplement your research through interviews of network members, asking them to identify any other individuals and organizations that should be included in the network.

Definitional Focus: Attributes, Relations, or Events

This approach to drawing boundaries leads researchers to focus on certain features of a network, namely, the attributes of actors, types of relations, or participation in events, while leaving the remaining features free to vary (Laumann et al. 1983:22). Social network analysts who focus on the *attributes* of actors generally do so in one of two ways: either in terms of position (i.e., where a membership test refers to the presence or absence of some attribute, such as holding a position in a formal group) or reputation (e.g., one that draws on the judgments of knowledgeable informants for identifying participant actors). Examples would include tracing the network of individuals who are members of a terrorist group (International Crisis Group 2006, 2009b) or of organizations that are involved in a particular social movement (Osa 2003).

When social network analysts use a relational focus to determining network boundaries, they focus on a specific type of tie (or set of ties) between actors (e.g., friendship, kinship, business, school, faith community). For example, researchers may be interested in studying the friendship and acquaintance ties of a particular high school or elementary school (McFarland 2004). If so, then they could (theoretically) obtain a roster of the students and then ask them to identify whom they consider to be their close friends, acquaintances, and so on. Krackhardt's (1987a, 1992) examination of the advice and friendship networks of a Silicon Valley high-technology firm is also an example of this approach. In the case of dark networks, analysts may be interested in mapping the trust network of a particular dark network and consequently focus on those types of ties that are indicative of trust (e.g., friendship, kinship, religious, school).

Finally, some researchers use participation in a particular event or activity to select actors and the social relationships among them into a network. The Southern Club Women network, which was constructed based on the attendance of eighteen women at fourteen different social events (Davis, Gardner, and Gardner 1941), is one of the more widely known examples of this approach. It is a favorite among social theorists (Breiger 1974; Homans 1950), and one of the standard datasets included with UCINET. Other examples of networks delineated based on this approach include the network of individuals who participated in the Madrid bombing (Rodriguez 2005), the first Bali bombing (Koschade 2006), and the network of nations active in the global economy (Smith and White 1992).

It is important to stress that these foci are not necessarily mutually exclusive, so we will sometimes want to use them in conjunction with one another. Indeed, it may be more common for researchers to use multiple foci rather than only one. For example, we may examine the friendship ties of members of a particular community of faith as well as ties formed through participation on various boards and committees.

Summary

Table 4.1 combines these two dimensions into a single matrix in order to illustrate the array of possible approaches that researchers can use to specify the boundaries of the network they are studying. What should be clear is that analysts are not limited to a single type of approach (i.e., types I through VI in the table) but rather can adopt approaches that combine two or more foci and land somewhere on the continuum between nominalist and realist strategies. What is ultimately important is not the approach taken but rather that the boundary of the network being analyzed is correctly specified. As with any empirical approach to studying social phenomena, misspecification can lead to erroneous conclusions, something we do not want to do when we are trying to track, disrupt, and/or destabilize a dark network.

Table 4.1. *Boundary specification typology*

	Definitional focus			
	Actor attributes	Type of relation	Event or activity	Multiple foci
Strategy	I	II	III	
Realist (Subjective)	• Actors included are members of a socially defined group that group members recognize • Example: Members of a self-identified high school group (e.g., jocks, stoners)	• Focus on members' degree or type of relation • Example: A primary (face-to-face) group or clique	• Actors included are those who attend or participate in a series of events or activities • Example: Davis et al.'s (1941) study of 18 women who attended 14 social events	• Combination of types I, II, and III
	IV	V	VI	
Nominalist (Objective)	• Focus on attributes objectively defined • Example: Business elite defined as members of Forbes 500 corporation boards	• Actors' inclusion based on their presence in a type of relation • Example: Krackhardt's (1987, 1992) Advice and Friendship Networks	• Difficult to define as participation in event is self-conscious activity • Example: Citation networks, "invisible" college of academics – membership determined by areas of interest (e.g., SNA)	• Combination of types IV, V, and VI

Note: Adapted from Laumann, Marsden, and Prensky (1983).

4.3 Ego Networks and Complete Networks

A difficulty facing social network analysts is that it is next to impossible to study very large social networks – for example, the United States, California, Silicon Valley, and San Jose. It would be wonderful if social network analysts could use sampling to collect social network data because then we could generalize our findings to entire populations. Unfortunately, sampling does not work for most forms of SNA, and it is easy to see why. Imagine if we drew a sample of 1,200 individuals

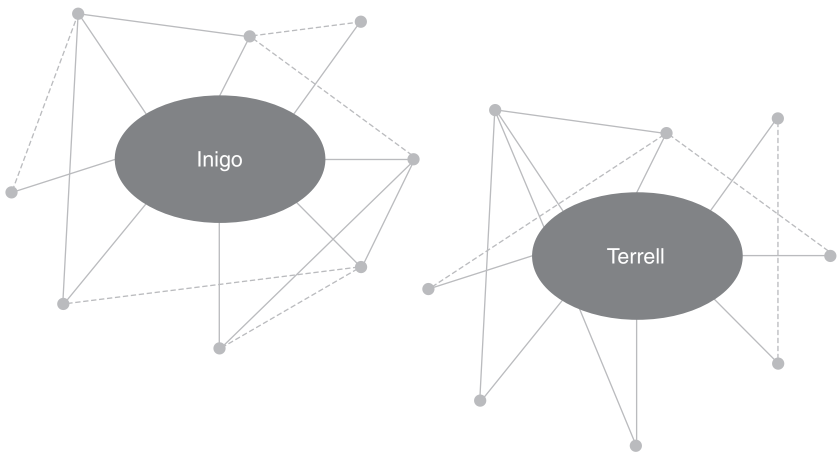


Figure 4.1. Hypothetical Ego Network

and asked them to name their friends and identify the ties between them. Chances are that their friends would not be part of the original sample, which means that we would not have enough relational data to analyze because most of the people surveyed would not know one another. Social network researchers have responded to this fact in two different ways. One approach actually uses sampling but focuses only on what social network analysts refer to as ego networks; the other, which is more common, analyzes what social network analysts call complete social networks. We briefly consider each of these.

Ego Networks

An ego-centered approach focuses on the person surveyed – typically termed *ego* – and the set of contacts (i.e., *alters*) who have ties to the person and measurements on the ties among these alters. Each person surveyed is generally asked for a set of contacts (Burt 1984, 1985), using questions such as "Looking back over the last six months, who are the people with whom you discussed matters important to you?" After providing a list of contacts, they are then asked about the ties (if any) between their contacts (e.g., if they know one another, attend the same church, are friends, and so on). Needless to say, ties between an ego and his or her alters with different egos and their alters are not (and generally cannot be) recorded. This yields a data structure similar to that displayed in Figure 4.1. As one can see, only those ties within the ego network of the people sampled (i.e., Inigo and Terrel) are recorded, whereas ties between ego networks are not.

A common use of ego network data is to estimate the size of peoples' core networks to see whether it has changed over time or is correlated with certain types of behavior (Marsden 1987). For example, Miller McPherson, Lynn Smith-Lovin, and Matthew Brashears (2006), using data gathered with the 2004 General Social Survey (GSS), concluded that from 1985 to 2004 the average size of individuals' core discussion networks dropped from 2.94 to 2.08, whereas the modal size dropped from 3.00 to 0.00. According to their study, almost one quarter of the population now reports that they do not discuss important matters with anyone! Their conclusions have been challenged by sociologist Claude Fischer who believes the research team's findings are highly implausible. He contends

> that the question used in the 2004 survey to measure the size of respondents' networks yielded results that were so inconsistent with other data, and so internally anomalous and implausible, that they are almost surely the product of an artifact. These data do not provide a reliable estimate of what happened to Americans' networks between the 1980s and 2004. (Fischer 2009: 657)

This led him to conclude that we should not infer from the 2004 GSS that Americans' social networks have changed substantially between 1985 and 2004. His conclusions also highlight the difficulty of generalizing ego network data gathered in surveys to the wider population.

Complete Networks

A more common approach to SNA collects relational data on an entire network (assuming that we know the boundaries of that network – see Section 4.2). Most social network methodologies are built on the assumption that the network being studied is a complete network that not only includes all relevant actors but also all relevant ties between actors. Because the complete network approach places limits on the size of the networks that can be studied, social network analysts focus primarily on case studies, which is what we will do in this book.

4.4 Types of Social Network Data

Social network analysts work with three types of data: one-mode social network data (symmetric and asymmetric), two-mode social network data, and attribute data. We discussed the nature of attribute data in Chapter 1, so here we limit our discussion to one- and two-mode network data.

	1 ACCIAIUOL	2 ALBIZZI	3 BARBADORI	4 BISCHERI	5 CASTELLAN	6 GINORI	7 GUADAGNI	8 LAMBERTES	9 MEDICI	10 PAZZI	11 PERUZZI	12 PUCCI	13 RIDOLFI	14 SALVIATI	15 STROZZI	16 TORNABUON
1 ACCIAIUOL	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
2 ALBIZZI	0	0	0	0	0	1	1	0	1	0	0	0	0	0	0	0
3 BARBADORI	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0
4 BISCHERI	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
5 CASTELLAN	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	0
6 GINORI	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7 GUADAGNI	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1
8 LAMBERTES	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
9 MEDICI	1	1	1	0	0	0	0	0	0	0	0	0	1	1	0	1
10 PAZZI	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
11 PERUZZI	0	0	0	1	1	0	0	0	0	0	0	0	0	0	1	0
12 PUCCI	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13 RIDOLFI	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	1
14 SALVIATI	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0
15 STROZZI	0	0	0	1	1	0	0	0	0	0	1	0	1	0	0	0
16 TORNABUON	0	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0

Figure 4.2. Subset of Padgett and Ansell's Marriage Data

Symmetric One-Mode Networks

One-mode networks, sometimes referred to as adjacency matrices, consist of a single set of actors, which can be people, groups, families, tribes, organizations, corporations, nation-states, etc. The ties between actors can be friendship or kinship ties; material transactions such as business transactions including the import or export of goods; communication networks involving the sending or receiving of messages; and so on. An example of a one-mode network, which we briefly examined in the Chapter 2, is Padgett and Ansell's (1993) Florentine families network. Padgett and Ansell collected nine types of relational data on ninety-two prominent fifteenth-century Florentine families in order to explain Cosimo de' Medici's rise to power. Included with UCINET is a subset of this data that delineates the business and marriage ties between sixteen of the ninety-two families (Breiger and Pattison 1986). A marital tie was determined to exist if a member of one family married a member of another family, while a business tie was determined to exist if a member of one family granted credits, made a loan, or entered into a joint partnership with a member of another family (Wasserman and Faust 1994). Figure 4.2 presents a UCINET display (i.e., *Data>Display*) of the Padgett and Ansell's marriage data. *Data>Display*

One-mode networks always result in square matrices because each actor (in this case, each family) appears as both a row and a column. For example, the Acciaiuol family's ties are recorded in both the first row and first column, the Albizzi family's ties are recorded in the second row and second column, the Barbadori family's ties in the third row and third column, and so on. In this case, the ties are *dichotomous* because they only take the values of "0" or "1," with "1" indicating the presence

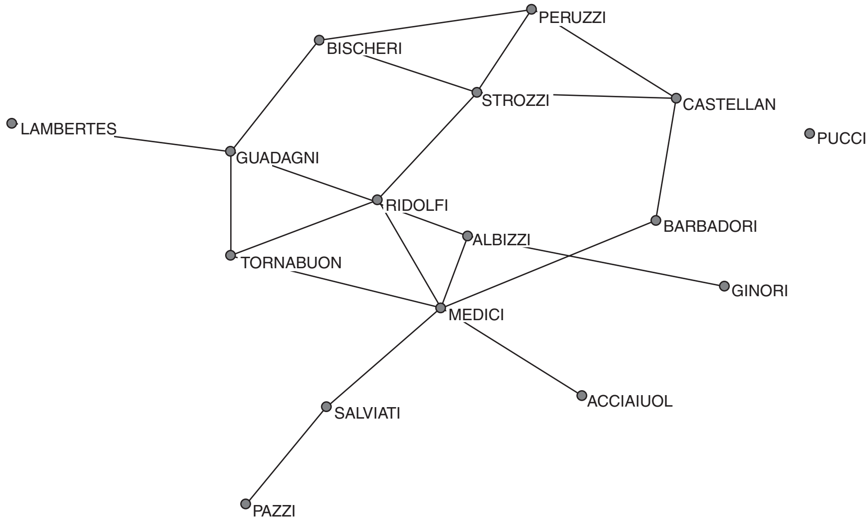


Figure 4.3. Sociogram of Padgett and Ansell's Marriage Data

of a marriage tie and "0" indicating the absence of one.[2] Also, the ties between families are necessarily reciprocal; they go both ways. In other words, not only does the Albizzi family have a marriage tie with the Ginori family (note the "1" in row 2, column 6), but the Ginori family has a marriage tie with the Albizzi family (note the "1" in row 6, column 2). From this matrix, you can see that the Medici family had ties to six families (Acciaiuol, Albizzi, Barbadori, Ridolfi, Salviati, and Tornabuon), the Strozzi family had ties to four families (Bishcheri, Castellan, Peruzzi, and Salviati), while the Pucci family has ties to no other family. Figure 4.3 displays a sociogram (i.e., network map) created in Pajek of the network. As you can see, the Pucci family is the only family that shares no marriage ties with any of the other families. The Medicis are also relatively central to the network although they do not appear to be any more central than some of the other families. Of course, this is just a subset of the data, so it may not be indicative of the actual Florentine families network.

Asymmetric One-Mode Networks

Ties are not always reciprocal, of course. Take, for example, the social network data we explored earlier, the data collected by David Krackhardt (1987a, 1992) on twenty-one managers in a Silicon Valley high-technology company. You will recall that Krackhardt asked each manager to whom they went to for advice and whom they considered a friend. He

[2] Cell values can also be "valued," indicating some sort of numerical relationship between two actors. For example, a cell may indicate the amount of imports between two countries.

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
		--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--
1	1	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	1	0	0	1
2	2	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
3	3	1	1	0	1	0	1	1	1	1	1	1	1	0	1	0	0	1	1	0	1	1
4	4	1	1	0	0	0	1	0	1	0	1	1	1	0	0	0	1	1	1	0	1	1
5	5	1	1	0	0	0	1	1	1	0	1	1	0	1	1	0	1	1	1	1	1	1
6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
7	7	0	1	0	0	0	1	0	0	0	0	1	1	0	1	0	0	1	1	0	0	1
8	8	0	1	0	1	0	1	1	0	0	1	1	0	0	0	0	0	0	1	0	0	1
9	9	1	1	0	0	0	1	1	1	0	1	1	1	0	1	0	1	1	1	0	0	1
10	10	1	1	1	1	1	0	0	1	0	0	1	0	1	0	1	1	1	1	1	1	0
11	11	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	12	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
13	13	1	1	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0
14	14	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1
15	15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1
16	16	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0
17	17	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
18	18	1	1	1	1	1	0	1	1	1	1	1	0	1	1	1	1	0	0	1	1	1
19	19	1	1	1	0	1	0	1	0	0	1	1	0	0	1	1	0	0	1	0	1	0
20	20	1	1	0	0	0	1	0	1	0	0	1	1	0	1	1	1	1	1	0	0	1
21	21	0	1	1	1	0	1	1	1	0	0	0	1	0	1	0	0	1	1	0	1	0

Figure 4.4. Krackhardt Advice Network Data

also determined from company documents to which manager each manager reported. Figure 4.4 presents the advice network data in matrix form, which indicates that although virtually every manager seeks advice from managers #2 and #21, managers #2 and #21 do not always reciprocate. For example, while manager #1 seeks advice from manager #21, manager #21 does not seek advice from manager #1 in return. By contrast, manager #15 is not too popular in terms of giving advice, but he is not shy in asking for it himself. While only four managers (10, 18, 19, and 20) seek his advice, he seeks advice from every other manager in the company.

Figure 4.5 presents a Pajek-generated sociogram of the network. While no major patterns jump out, at first glance some managers appear more

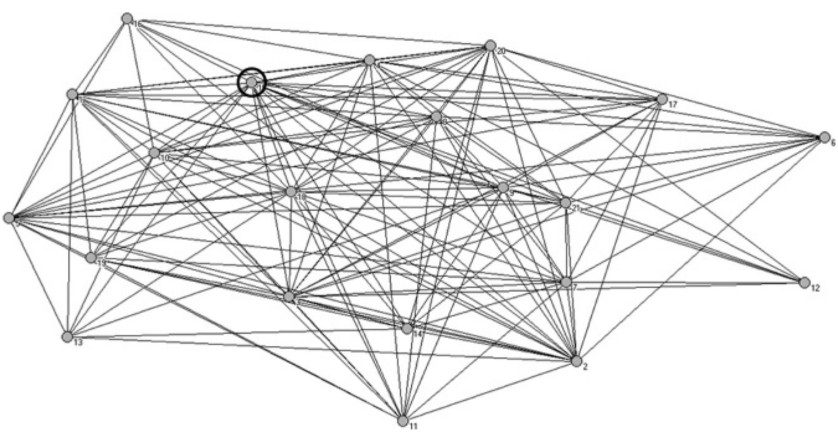


Figure 4.5. Sociogram of Krackhardt's Advice Network

		1 E1	2 E2	3 E3	4 E4	5 E5	6 E6	7 E7	8 E8	9 E9	10 E10	11 E11	12 E12	13 E13	14 E14
1	EVELYN	1	1	1	1	1	1	0	1	1	0	0	0	0	0
2	LAURA	1	1	1	0	1	1	1	1	0	0	0	0	0	0
3	THERESA	0	1	1	1	1	1	1	1	1	0	0	0	0	0
4	BRENDA	1	0	1	1	1	1	1	1	0	0	0	0	0	0
5	CHARLOTTE	0	0	1	1	1	0	1	0	0	0	0	0	0	0
6	FRANCES	0	0	1	0	1	1	0	1	0	0	0	0	0	0
7	ELEANOR	0	0	0	0	1	1	1	1	0	0	0	0	0	0
8	PEARL	0	0	0	0	0	1	0	1	1	0	0	0	0	0
9	RUTH	0	0	0	0	1	0	1	1	1	0	0	0	0	0
10	VERNE	0	0	0	0	0	0	1	1	1	0	0	1	0	0
11	MYRNA	0	0	0	0	0	0	0	1	1	1	0	1	0	0
12	KATHERINE	0	0	0	0	0	0	0	1	1	1	0	1	1	1
13	SYLVIA	0	0	0	0	0	0	1	1	1	1	0	1	1	1
14	NORA	0	0	0	0	0	1	1	0	1	1	1	1	1	1
15	HELEN	0	0	0	0	0	0	1	1	0	1	1	1	0	0
16	DOROTHY	0	0	0	0	0	0	0	1	1	0	0	0	0	0
17	OLIVIA	0	0	0	0	0	0	0	0	1	0	1	0	0	0
18	FLORA	0	0	0	0	0	0	0	0	1	0	1	0	0	0

Figure 4.6. Davis's Southern Women Network Data

central than others. The centrality of some may be misleading, however. Take manager #15, for example (circled in Figure 4.5). He is relatively central in the sociogram, but the reason is because, as we discussed previously, he indicated that he sought advice from every other manager in the company, which means that he has ties to every other manager, why he appears to be a central player in the network, and why we cannot rely on network visualizations alone.

Two-Mode Networks

Two-mode networks differ from one-mode networks in that rather than consisting of a single set of actors, they either consist of two sets of different actors, or one set of actors and one set of events or affiliations. Researchers often refer to two-mode networks as *affiliation matrices* or *networks*, but they sometimes call them *membership networks*, *dual networks*, and *hypernetworks*. Examples of two-mode networks include membership in various organizations, attendance at particular events, employees at a particular company, and so on. An example of a two-mode network is Davis's Southern Women, which we discussed in Section 4.2 (Breiger 1974; Davis, Gardner, and Gardner 1941) and records the observed attendance of eighteen southern women at fourteen social events. The women are listed by row; the events by column. As Figure 4.6 indicates, Evelyn attended eight events (1, 2, 3, 4, 5, 6, 8, and 9), while Olivia, Dorothy, and Flora attended only two (9 and 11).

A key assumption underlying the use of two-mode networks by social network analysts is that membership in an organization or participation in an event is a source of social ties. Why? Because people who join or participate in a common organization and/or event often share similar tasks and/or interests, and they are much more likely to interact with one another than two randomly selected people. That said, we need to be careful when using two-mode data. Just because two people participate in a common event or are members of the same faith community does not necessarily mean that a tie exists between them.

A Pajek-generated sociogram of Davis's Southern Women appears in Figure 4.7. Note the difference between this one and the previous two. Here, both sets of actors, the women and events, appear together and are assigned different colors (or, in this case, shades of gray). The graph suggests that there are three clusters of events (E1–E5, E6–E9, and E10–E14) and two primary sets of women (Laura through Ruth; Silvia through Verne), plus a few women who appear less involved than the rest (Dorothy, Olivia, and Flora). As we will see, there are ways of converting two-mode networks to one-mode networks, which is another way of exploring which actors are central and which ones are not.

4.5 Collecting Social Network Data

Social network analysts collect social network data in a variety of ways. The most common are questionnaires, interviews, direct observation, and written records (Wasserman and Faust 1994:45–54). Not all of these are useful for collecting data on dark networks, but it is still useful to briefly consider the various approaches for collecting social network data.[3]

Questionnaires

Questionnaires are a common method for collecting social network data, especially when actors are individuals, although they are probably the least common method when it comes to collecting relational data on dark networks. They contain questions such as whom people consider to be their friends, to whom do they go to for advice, with whom do they regularly communicate (e.g., talk face-to-face, e-mail, telephone), and so on. Such data can be recorded either symmetrically or asymmetrically. Say, for instance, actor "A" considers actor "B" to be a friend, but "B" does not consider "A" to be a friend. In such a case, researchers can either record the data as Krackhardt did (i.e., asymmetrically) by placing a "1" in the "A-B" cell of the matrix but a "0" in the "B-A" cell of the matrix, or they can record it symmetrically by placing a "0" in both cells under the assumption that a friendship tie only exists if both actors indicate that they consider the other to be a friend. Analysts use various formats for collecting social network data using questionnaires; they fall under three broad categories: (1) Roster vs. Free Recall, (2) Free vs. Fixed Choice, and (3) Ratings vs. Complete Rankings (Wasserman and Faust 1994:45).

Roster vs. Free Recall. Sometimes analysts present each respondent filling out a questionnaire with a complete roster of the actors in the network

[3] For a more in-depth but relatively brief summary of various methods for collecting social network data, see Prell (2011:68–74).

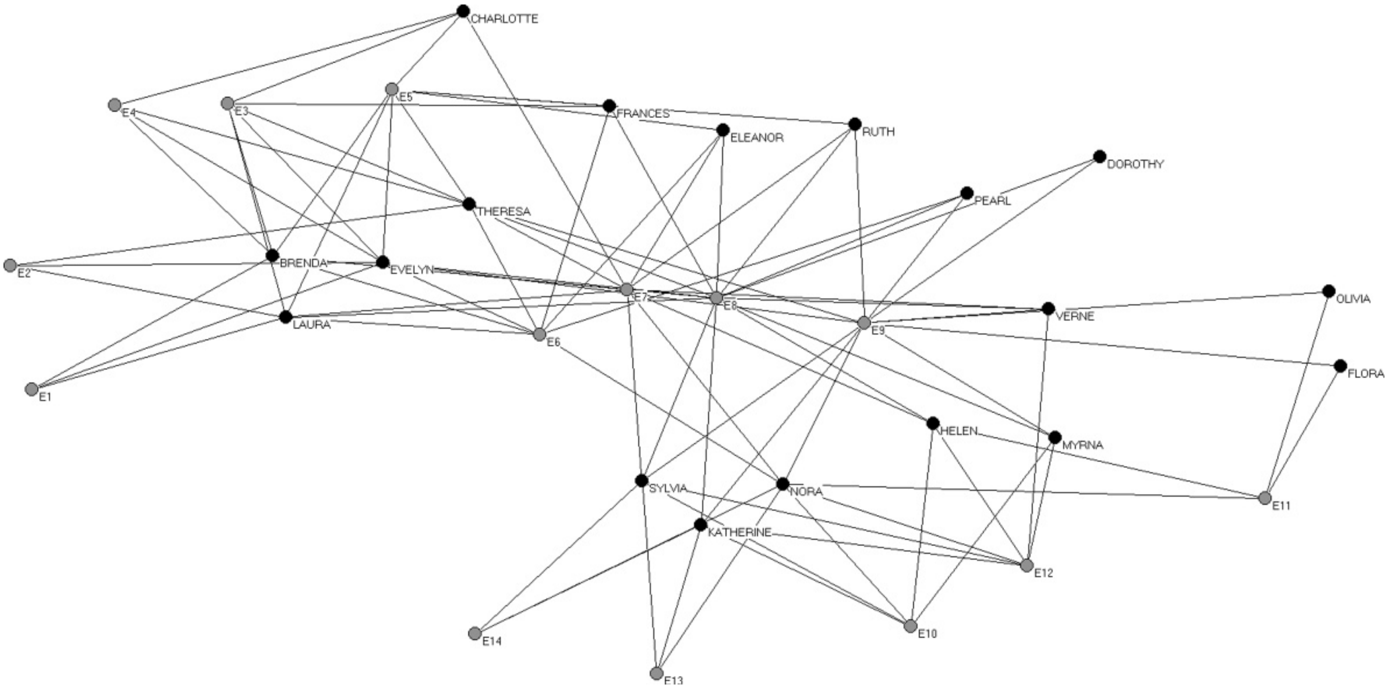


Figure 4.7. Sociogram of Davis's Southern Women

or allow the respondents to generate a list of names. Rosters can only be used when researchers know the members in the network prior to gathering data (Wasserman and Faust 1994:46). This, of course, raises the network boundary issue discussed earlier: How do researchers know a priori which actors belong to a network and which ones do not? When working with a self-contained organization (e.g., a small high-technology start-up), this is sometimes relatively obvious (at least for the purposes of the study), but it is not always so clear-cut. In the latter case, it is usually advisable to use the free-recall approach.

Free vs. Fixed Choice. In some network designs analysts tell respondents how many other actors they are to nominate on a questionnaire (e.g., "Name five people with whom you have regular contact"); at other times they are not presented with any such constraints as to how many nominations they can make (e.g., "Name everyone with whom you have regular contact"). The former (fixed choice) can underestimate the size or density of a network and produce misleading results.

Ratings vs. Complete Rankings. Finally, sometimes analysts ask each respondent to rate or rank the ties in terms of strength between all actors in the network (Wasserman and Faust 1994:48). Ratings can be either dichotomous (e.g., ties are either present or absent) or valued (e.g., respondents choose one of a few possible categories for the strength of each tie). Rankings differ in that each actor is asked to rank their ties to every other actor in the network. This latter approach becomes increasingly difficult as the size of the network increases.

Interviews

Social network analysts sometimes use interviews (either face to face or over the phone; Wasserman and Faust 1994:48). Interviews with captured members of dark networks may prove useful for mapping the networks, but these should probably be supplemented with other methods (e.g., direct observation, written records).

Direct Observation

Another way to record data is to have an observer record all interactions that take place among actors in the network (Wasserman and Faust 1994:49). Dan McFarland (2004) used this approach to record student interaction patterns at two different high schools. An obvious drawback to this approach is that in some situations interactions can be so numerous and occur so closely together that it becomes next to impossible to record all interactions. Moreover, those being observed often alter their behavior when they are aware that their interactions are being recorded (Roethlisberger and Dickson 1939). Nevertheless, analysts of dark networks might find this approach useful when recording affiliation (i.e., two-mode) network data. For example, they can record which members of a particular dark network visit specific sites or attend specific events in connection with their participation with the network.

Written Records

Written records can be valuable sources of relational data. E-mails, memos, phone calls (if available), historical marriage records, and court proceedings are just a few examples of sources from which one can determine ties between individual actors. At the corporate level, written records indicating joint ventures, interlocking directorates (i.e., where the same individual sits on the boards of two different companies), and membership in the same trade association may indicate ties, while records indicating the trade manufactured goods or the exchange of diplomats may indicate ties between countries. In terms of collecting relational data on dark networks, Sageman (2004, 2008:26–27) drew on captured documents, trial transcripts, intercepted conversations, legal documents, and testimony notes in order to determine some of the ties among members of the global salafi jihad. And throughout this book we draw on a narrative about Noordin's terrorist network that, in turn, drew in part on court records (International Crisis Group 2006).

Other Approaches

These are not the only approaches to collecting social network data. They are simply the most common. Other forms of data collection include cognitive social structure data, experiments, diaries, and small worlds (Wasserman and Faust 1994:51–54). When collecting cognitive social structure data, researchers ask respondents for their perception of other actors' network ties (e.g., "Who is friends with whom?"; Krackhardt 1987a). Social network analysts sometimes use experiments to observe the behavior of a set of actors in experimentally controlled environments (Bavelas 1950; Emerson 1962). Diaries are used by social network analysts to ask respondents to keep a continuous record of people with whom they interact (Wasserman and Faust 1994:54). And finally, researchers will use variations on small world network design (Milgram 1967; Travers and Milgram 1969) to estimate how many steps (i.e., degrees of separation) a respondent is removed from a randomly chosen target (Watts, Dodds, and Newman 2003).

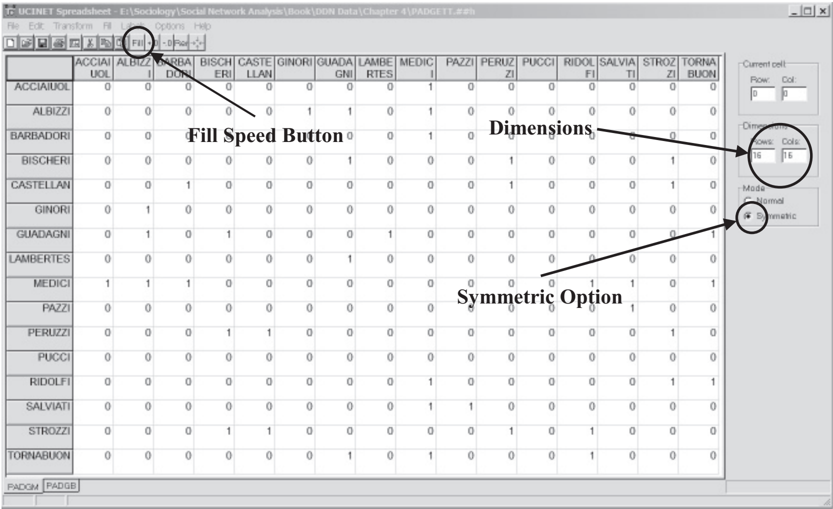


Figure 4.8. Padgett and Ansell's Marriage Network (UCINET)

4.6 Recording Social Network Data

In this section we illustrate how to record social network data in both UCINET and ORA and then export these data in formats readable by Pajek.[4] We begin by demonstrating the process for recording symmetric and asymmetric one-mode networks before moving to demonstrating it for a two-mode social network. Because so many network datasets often come in UCINET format, we also show how to read such datasets in Pajek and ORA. This section also shows (and briefly discusses) the recording of attribute data. The next section takes up how to aggregate and parse multiple networks.

Recording One-Mode Social Network Data

UCINET: Recording One-Mode Social Network Data. In UCINET, social network data can easily be recorded in matrix form using its internal spreadsheet editor. For example, if we wanted to enter Padgett marriage data discussed previously, we would first open the spreadsheet editor (Figure 4.8) using either the *Data>Data editors>Matrix Editor* command or the "Matrix" speed button located just under the *Data* menu (see Figure 3.1 in Chapter 3).

[UCINET] Data>Data editors >Matrix Editor

Next, because we are entering symmetric network data, we should select the "symmetric" option located on the right panel of the spreadsheet

[4] Although you can record network data using Pajek, it is generally easier to record the data in either UCINET or ORA and then export the data in a form that Pajek can read.

editor (circled in Figure 4.8). This option ensures that each entry we make is replicated in the corresponding cell. In other words, if we enter a "1" in the cell located in the third row and the first column, UCINET automatically enters a "1" in the cell located in the first row and third column. Before entering values into cells, however, enter the names of the families in the rows (UCINET's symmetric option will automatically place them in the corresponding column). Then enter "1s" wherever a tie between two families exists (see Figure 4.8). UCINET's "Fill" feature means that we do not have to enter "0s." Instead, after indicating the dimensions of the matrix (i.e., the number of rows and columns) on the right side of the spreadsheet editor, use either the *Fill>Blanks w/0s* command or click the "Fill" speed button to fill all the empty cells with "0s." Your completed spreadsheet should look similar to Figure 4.8.

[Spreadsheet] Fill>Blanks w/0s

Of course, if we are entering asymmetric network data, we would not want to select the symmetric option. There is no point in illustrating this here, because, except for selecting the symmetric option, the process of entering asymmetric network data is identical to that of symmetric data. Instead, we will discuss how to enter social network data in ORA using an asymmetric network.

ORA: Recording One-Mode Social Network Data. We will use Krackhardt's advice network to illustrate how to record one-mode social network data in ORA. The first step in the process is to access ORA's "Create New Meta-Network" dialog box with the *Generate Networks>Create New Meta-Network* command. Next, specify the name of the meta-network and then indicate the various types of actors (i.e., node classes) in the network. This is a one-mode network, by definition there is only one type of actor, and in this instance they are individuals, so we should indicate that both types of nodes are agents. We also need to tell ORA that the size of the network (i.e., node class) is 21 (i.e., 21 managers).

[ORA] Generate Networks >Create New Meta-Network

Finally, we have to identify the various types of networks that will be included in the meta-network – advice, friendship, and reports to – and that they are all one-mode networks (i.e., source and target actor/node type are the same). After all this is entered (Figure 4.9), click "Create" and a new meta-network (without the ties) will be created.

The next step involves recording the ties between actors. Select the advice network in the Meta-Network Manager and then click on the Editor tab in the Network Information/Editor panel. Because this is a binary (i.e., dichotomous) network, we simply check the cells where a tie exists between two actors. In ORA we do not need to record "0s" because ORA assumes that a cell value is "0" unless it is checked. When you are finished, it should look similar to Figure 4.10. If we want to record valued data (as opposed to just "1s" and "0s"), then we would click and

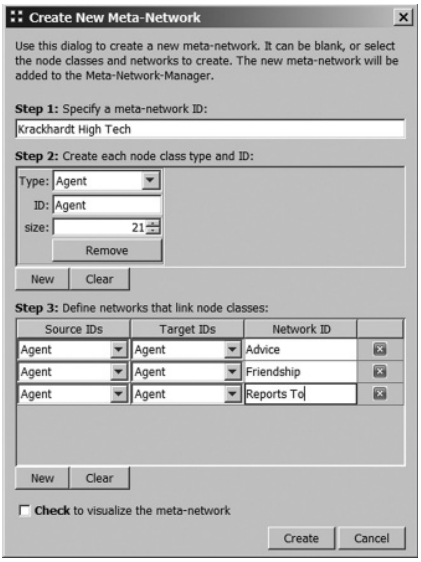


Figure 4.9. ORA's New Meta-Network Dialog Box (Krackhardt High-Tech)

hold down the "Display Options" button in the lower-right portion of the editor and select the "Numeric Link Values" option.

Recording Two-Mode Social Network Data

UCINET: Recording Two-Mode Social Network Data. We enter two-mode data into UCINET's spreadsheet function similar to the way we did

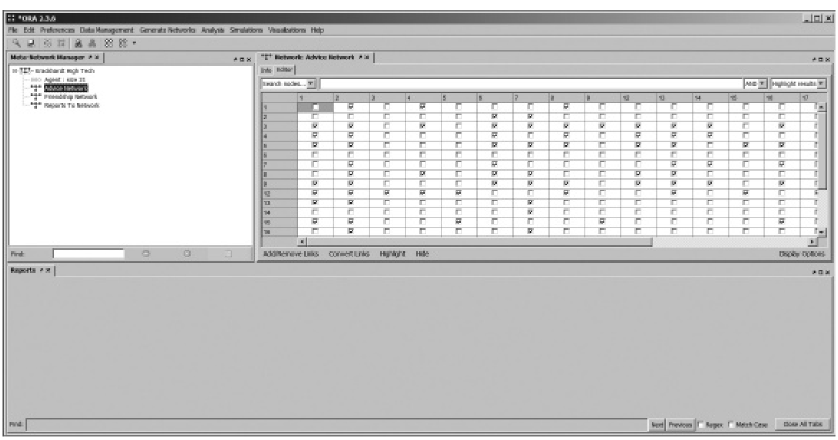
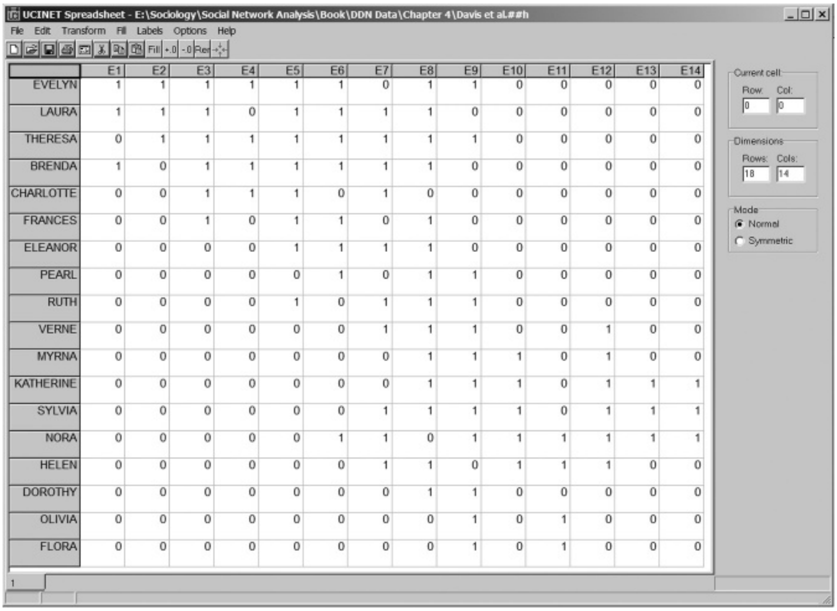


Figure 4.10. Krackhardt's Advice Network Matrix (ORA)



UCINET Spreadsheet - E:\Sociology\Social Network Analysis\Book\DDN Data\Chapter 4\Davis et al.##h

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14
EVELYN	1	1	1	1	1	1	0	1	1	0	0	0	0	0
LAURA	1	1	1	0	1	1	1	1	0	0	0	0	0	0
THERESA	0	1	1	1	1	1	1	1	1	0	0	0	0	0
BRENDA	1	0	1	1	1	1	1	1	0	0	0	0	0	0
CHARLOTTE	0	0	1	1	1	0	1	0	0	0	0	0	0	0
FRANCES	0	0	1	0	1	1	0	1	0	0	0	0	0	0
ELEANOR	0	0	0	0	1	1	1	1	0	0	0	0	0	0
PEARL	0	0	0	0	0	1	0	1	1	0	0	0	0	0
RUTH	0	0	0	0	1	0	1	1	1	0	0	0	0	0
VERNE	0	0	0	0	0	0	1	1	1	0	0	1	0	0
MYRNA	0	0	0	0	0	0	0	1	1	1	0	1	0	0
KATHERINE	0	0	0	0	0	0	0	1	1	1	0	1	1	1
SYLVIA	0	0	0	0	0	0	1	1	1	1	0	1	1	1
NORA	0	0	0	0	0	1	1	0	1	1	1	1	1	1
HELEN	0	0	0	0	0	0	1	1	0	1	1	1	0	0
DOROTHY	0	0	0	0	0	0	0	1	1	0	0	0	0	0
OLIVIA	0	0	0	0	0	0	0	0	1	0	1	0	0	0
FLORA	0	0	0	0	0	0	0	0	1	0	1	0	0	0

Figure 4.11. Davis's Southern Women Matrix (UCINET Spreadsheet Editor)

with one-mode data, except that we do not select the symmetric option and the form of data entry differs because rather than consisting of a single set of actors, two-mode networks either consist of two sets of different actors, or one set of actors and one set of events or affiliations. Taking Davis's Southern Women as an example: The names of the women appear in rows, whereas the columns list the various events that the women attended (Figure 4.11). This could easily be reversed where the names of the events appear in rows, and the names of the women appear in the columns. However, social network analysts often work with an implicit "left to right" logic. Thus, because the women are more "logically" seen as attending various events (rather than the events attracting the women – although this is true as well), the women appear first (in rows), and the events appear second (in columns). Here, we can see that Evelyn attended events E1 through E6, E8, and E9, while Laura attended events E1 through E3 and E5 through E7. The same logic holds when recording actors' membership in various institutions. For instance, if a series of actors attended one (or more) faith-based organizations (e.g., churches, synagogues, temples, mosques), then we would typically list the actors first (i.e., in the rows) and the organizations second (i.e., in the columns).

ORA: Recording Two-Mode Social Network Data. Entering two-mode social network data in ORA is quite similar to entering one-mode data,

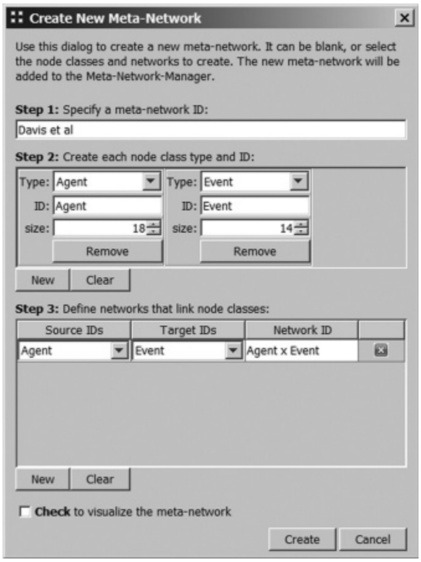


Figure 4.12. ORA's Create New Meta-Network Dialog Box (Davis's Southern Women)

although there are differences. We begin as we did before with ORA's *Generate Networks>Create New Meta-Network* command. Next, we specify the name of the meta-network and delineate the various types of actors/nodes (i.e., node classes) in the network. Since this is a two-mode network, we need to indicate as such in the dialog box. As you can see in Figure 4.12, two types of node classes have been identified: agent and event; the first has 18 nodes/actors and the second has 14 nodes/actors. And, as before, we need to tell ORA the types of networks that will be in the meta-network; in this case, just one. Because this is two-mode network, the source and target node/actor types differ. The source type is an agent (i.e., the women); the target type is an event (i.e., the social events). After this information is entered, click "Create" and a new meta-network matrix is created into which we can enter the ties indicating which women attended which events. The process of entering these ties is identical to that of one-mode networks. Since they are binary, all we need to do is check the cells where ties exist between the women and the various events. The final matrix should look similar (but probably not identical) to Figure 4.13.

[ORA] Generate Networks >Create New Meta-Network

Recording Attribute Data

As we discussed earlier, although SNA focuses primarily on the pattern of ties between actors, it does not ignore actors' attribute data. We can

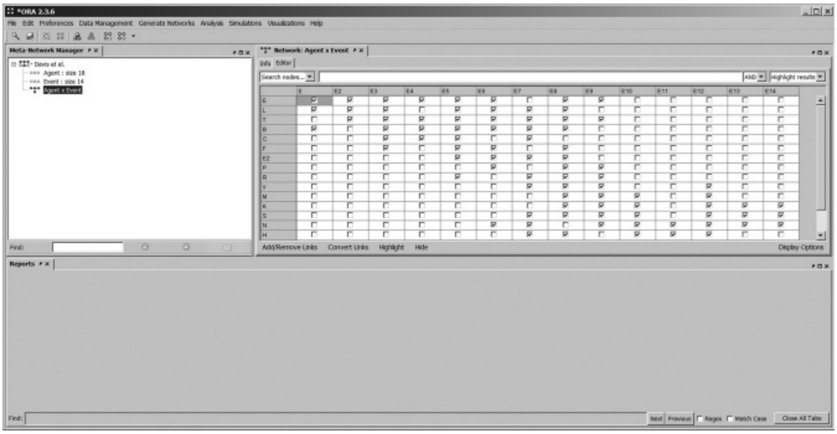


Figure 4.13. Davis's Southern Women Matrix (ORA)

use attribute data in various ways, some of which we will explore further on in this book. For now, it is only necessary to demonstrate how it is recorded in UCINET and ORA. As with social network data, if you want to use attribute data in Pajek, it is generally easier to record attribute data in either ORA or UCINET and then export it in formats that Pajek can read.

UCINET: Recording Attribute Data. Recording attribute data in UCINET is relatively straightforward. Figure 4.14 displays attribute data associated with Krackhardt's High-Tech network data. As you can see, the names of actors appear in rows, whereas the various types of attributes appear in columns. These data differ from social network data in that each column is self-contained. For example, the first column indicates the age of the managers while the fourth indicates the department to which each manager belongs. Obviously, without a code book, there is no way to know to what department each number refers.

ORA: Recording Attribute Data. In ORA, attribute data is recorded and stored in conjunction with a particular node class. To record attribute data in ORA, first select the "Agent" node class in the Meta-Network Manager panel and then click on the Editor tab in the Network Information/Editor panel (see Figure 4.15). On the far right of the screen, under the Attributes heading, note that there are a series of speed buttons for creating, importing, exporting, and deleting attribute data (we will consider the "Measures" option at a later time). To record attribute data, click on the "Create" button, which brings up a dialog box that asks you to provide a name for the attribute and to indicate what type it is. If you are entering nonordered data, then you will want to use either

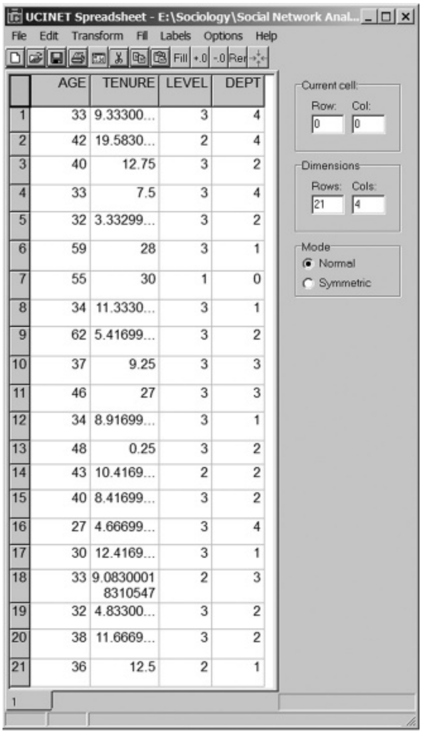


Figure 4.14. Krackhardt High-Tech Attribute Data (UCINET)

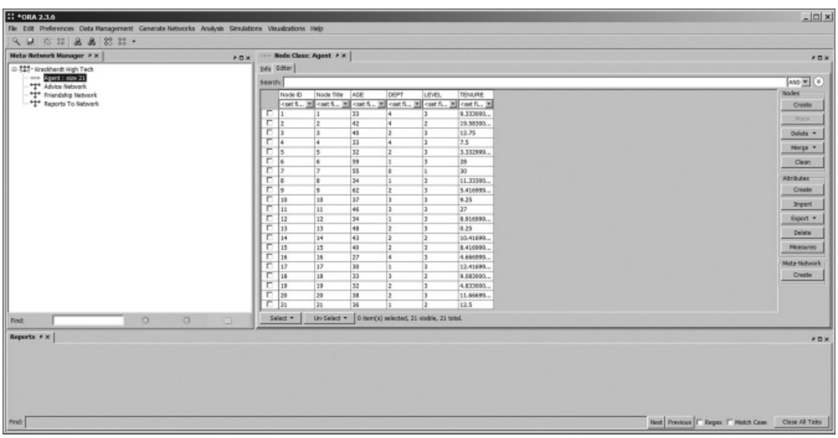


Figure 4.15. Krackhardt High-Tech Attribute Data (ORA)

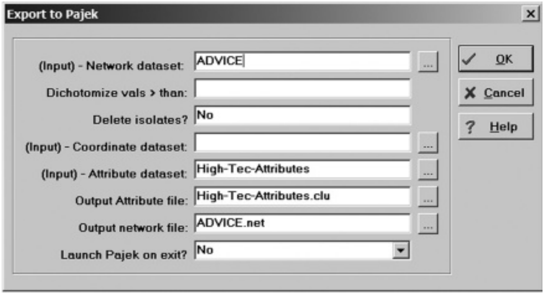


Figure 4.16. UCINET Export to Pajek Dialog Box

the "Text Category" or "Number Category" options; if you are entering ordered or continuous data, then you will want to use either the "Text" or "Number" options. Since the Krackhardt data are generally recorded as numbers, here we would select the "Number" option for entering the age, level, and tenure data, and the "Number Category" option for entering the department data.

Exporting Social Network Data to Pajek

[UCINET] Data >Export >Pajek >Network

UCINET: Exporting One-Mode Social Network Data. There are multiple methods for exporting one-mode social network data from UCINET to Pajek. The easiest way to export one-mode data is by using the *Data>Export>Pajek>Network* command (note that UCINET provides a choice of exporting the data in a number of formats: DL, Krackplot, Mage, Pajek, Metis, Raw, and Excel). This brings a dialog box (Figure 4.16). UCINET provides a series of options although most of the time you will want to accept UCINET's default settings. A good rule of thumb is that if you do not know what the option is asking, then accept UCINET's defaults (note that you can click on the dialog box's "Help" button for a detailed discussion of each of the options).

The one possible exception to this rule concerns the final option, which allows you to launch Pajek from within UCINET once the data are exported. You only want to use this option if (1) you do not already have Pajek open and (2) you have set UCINET's options so that it will open the most recent version of Pajek. Otherwise, it will launch the version of Pajek that comes with UCINET, which is often somewhat dated. If you choose yes, another dialog box will appear asking you (again) whether you want to launch Pajek. If all goes well, UCINET launches Pajek when you click "OK." If not, you can open Pajek and import the newly created Pajek network file manually.

Note that here, rather than exporting the entire Krackhardt HighTech file, we have only loaded the advice network to be exported. This is

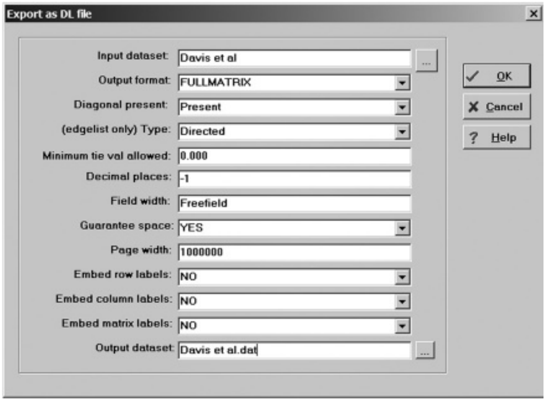


Figure 4.17. UCINET DL Export Dialog Box

because currently this command only exports single, one-mode networks and not datasets that include more than one one-mode network. If we wanted to export the entire Krackhardt file, we would need to use the same function we use to export two-mode networks from UCINET to Pajek: namely, UCINET's *Data>Export>DL* command (see the next section).

Data >Export >DL

Also note that UCINET allows us to export attribute data as well. Here, you can see that the attribute file associated with the Krackhardt data have been loaded for export. The one drawback to this function is that it exports all of the attributes as Pajek partition files, which if you recall, are designed for nonordered attribute data. Thus, if UCINET attribute files include continuous or ordered data (as these do), then we should export them as Pajek vector files. Luckily, UCINET includes commands for this. To export attribute data as Pajek partition files, use the *Data>Export>Pajek>Categorical Attribute* command; for Pajek vector files, use the *Data>Export>Pajek>Quantitative Attribute* command.

Data >Export >Pajek >Categorical Attribute, Quantitative Attribute

UCINET: Exporting Two-Mode Social Network Data. Because UCINET does not currently export two-mode networks in Pajek format, we have to export using its *Data>Export>DL* command. This calls up a dialog box (see Figure 4.17) where, once again, several options are offered but you will generally want to accept UCINET's defaults except the last one: Manually change the extension of the output dataset to "*.dat" (rather than "*.txt") because Pajek looks for DL files with *.dat extensions not *.txt ones.

Data >Export >DL

ORA: Exporting Social Network Data. Exporting network data from ORA to other programs is straightforward using ORA's *File>Data>Export* command. This brings up a dialog box (not shown) where you

[ORA] File>Data >Export

first use a drop-down menu to indicate which network you intend to export, then using a second drop-down menu you identify what type of format you wish to export the data. ORA offers several options: UCINET (typically, you will want to use the binary option), NetDraw, Pajek, and others. After selecting the type, click on the "Browse" button to either type in or locate the name of the file you are exporting. Then, select "Export." If things work as they should, your data will be waiting for you in the file format of your choice.

Importing/Reading UCINET Data into Pajek and ORA

Because so many network datasets often come in UCINET format, it is also helpful to demonstrate how to read or import such datasets into Pajek and ORA. We begin with Pajek before moving on to ORA.[5]

Pajek: Importing UCINET Data. As noted previously, in order to move social network data from UCINET to Pajek, we first need to export the data from UCINET in either Pajek or DL formats, the former for single, one-mode networks, the latter for multiple one-mode networks or single two-mode networks. To read either type of file into Pajek, we use Pajek's *File>Network>Read* command, which brings up a dialog box (not shown) that usually defaults to looking for Pajek-formatted files (i.e., network files with a *.net extension). Thus, we should not have to change any defaults if we are looking for Krackhardt's advice network, which we previously exported as a Pajek-formatted file. However, if we are looking for Davis's Southern Women file, which we exported as a DL-formatted file, then in the type of file drop-down menu, we need to indicate that we are looking for UCINET DL files (*.dat). Either way, select the file you want to import and click "Open."

[Pajek] File>Network >Read

One thing to keep in mind when importing data that originated as a UCINET file, regardless of whether you exported the data using UCINET's *Data>Export>Pajek>Network* command or its *Data> Export>DL* command, when Pajek reads the file into memory, it reads the ties as arcs (not edges). This is acceptable when you are working with directional data (i.e., asymmetric networks and two-mode networks), but if you are working with nondirectional data (i.e., symmetric networks), you will want to transform the arcs into edges using Pajek's *Net>Transform>Arcs➔Edges>All* command. This command calls up a dialog box (not shown) that asks if you want to create a new network. This is a good idea since you typically do not overwrite the original. Clicking "OK" brings up another dialog box (not shown), asking if you want to remove multiple lines. Select option five (single line) and click "OK."

Net >Transform >Arcs ➔Edges >All

5 As we saw in the Chapter 3, NetDraw reads UCINET files.

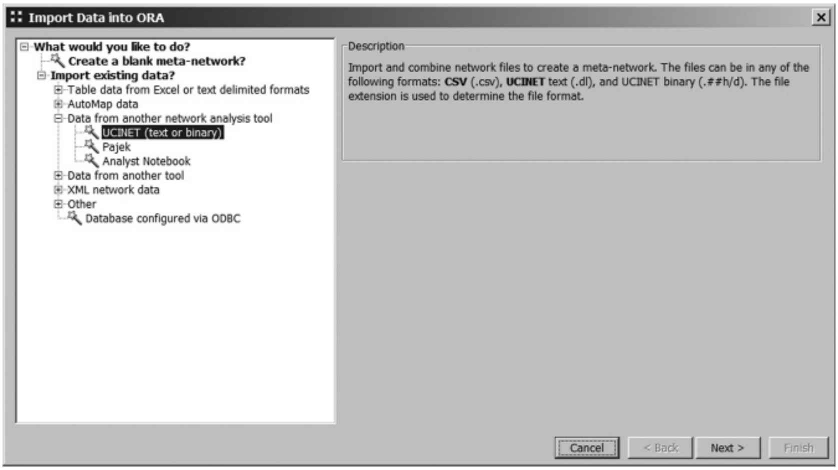


Figure 4.18. ORA's Data Import Wizard

You should now have a network that consists only of edges, not arcs. Importing attribute data originally created in UCINET is simple as well. To read nonordered attribute data, use Pajek's *File>Partition>Read* command. To read ordered or continuous attribute data, use Pajek's *File>Vector>Read* command.

File >Partition >Read

File>Vector >Read

ORA: Importing UCINET Data. To import network data into ORA created in another social network tool such as UCINET, we use the *File>Data Import Wizard*, which calls up a dialog box similar to Figure 4.18. Here, we have selected the "Data from another network analysis tool" option and highlighted "UCINET" as the type of network data we are importing (note that Pajek and Analyst Notebook are the other options). Clicking on the "Next" button brings up second dialog box (not shown) where you assign the network a name; clicking on "Next" calls up a third "Import Data into ORA" dialog box similar to Figure 4.19.

[ORA] File >Data Import Wizard

Using the "Browse" radio buttons (see Figure 4.19), locate the UCINET files you intend to import (in this case, it is the Krackhardt advice, friendship, and reports to networks–`ADVICE.##h, FRIENDSHIP.##h, REPORTS_TO.##h`), indicate that both its source and target type is "Agent," provide it with a name (i.e., Network ID), and click on the "Click to import another file" button. Repeat this process for each file you intend to import into the current meta-network. Here, we have also included the friendship and reports-to networks. Click "Finish," and you should be returned to ORA's main screen where the imported network is shown in the Meta-Network Manager panel.

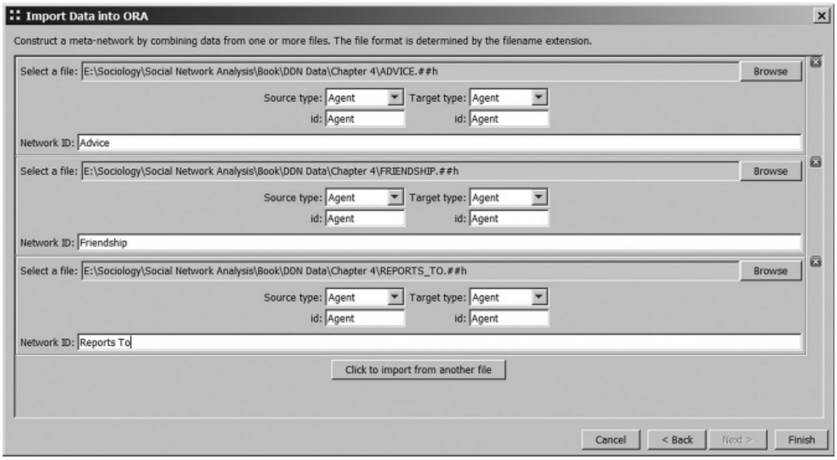


Figure 4.19. ORA's Data Import Dialog Box

To import attribute data into ORA, select the "Agent" node class in the Meta-Network Manager panel and click on the Editor tab in the Network Information/Editor panel (Figure 4.20). As we saw before, on the far right of the Editor tab, there are a series of speed buttons related to attribute data. Clicking on the "Import" button calls up a dialog box similar to Figure 4.21.

Currently, ORA's attribute import function reads `*.csv` files more reliably than *.xls files, so if your data are stored in an Excel file, you may need to first save your data as a *.csv file.[6] In this case, we have already done that for you. Use the "Browse" button to locate and select the attribute file you want to import. *It is imperative that it relates to the network file already loaded into ORA's memory*. Here, we are importing the `High Tech Attribute.csv.` Next, indicate that the rows are in the same order as are the network data. Then, uncheck the first box and then tell ORA that age, tenure, and level are "number" types of data, while department is either a text or number category type of data (see Figure 4.21). Finally, click on the "Import" button and the attribute data should be loaded into ORA.

4.7 Deriving One-Mode Networks from Two-Mode Networks

We can derive two one-mode networks (i.e., an actor-by-actor – "Co-membership" – network and an event-by-event – "event overlap" – network) from a two-mode network by multiplying the original affiliation

[6] You can also import attribute data in ORA using the *File> Data Import Wizard* function and assigning it to the appropriate node class during the process.

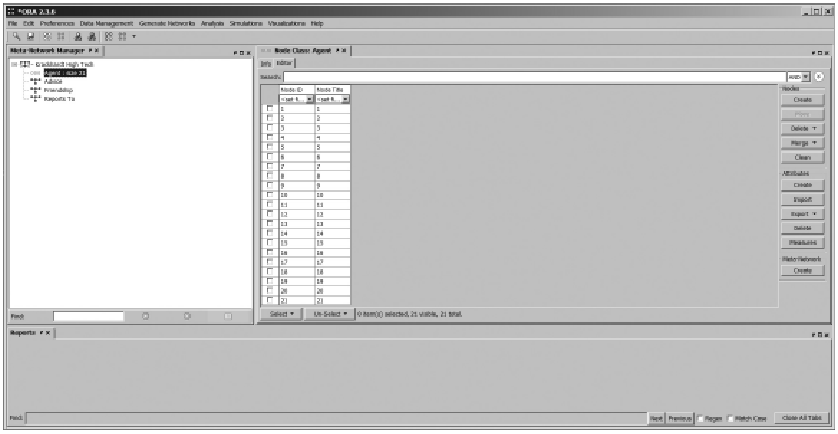


Figure 4.20. Importing Attribute Data in ORA

matrix by its transpose. Thankfully, UCINET, Pajek, and ORA have made this process relatively simple for users. Before turning to how to do this, we will first look at some of the interesting properties that such derived networks possess (Breiger 1974). Figure 4.22 displays co-membership network of Davis's Southern Women, and like all one-mode networks, the rows and columns refer to the same actors (in this case, the women who attended the social events). If two women attended the same event, then there will be a tie in the corresponding matrix cells (one-mode networks derived from two-mode networks are necessarily symmetric). If they attended more than one event together, then the value in the cell indicates the number. For example, Evelyn and Laura attended six of the same events, while Evelyn and Flora attended only one. Moreover, Evelyn

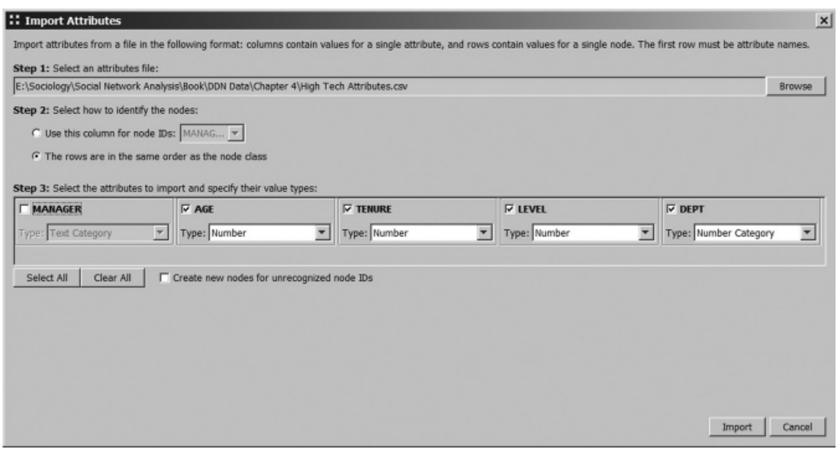


Figure 4.21. Importing Attribute Data in ORA Dialog Box

		1 E	2 L	3 T	4 B	5 C	6 F	7 E	8 P	9 R	10 V	11 M	12 K	13 S	14 N	15 H	16 D	17 O	18 F
1	EVELYN	8	6	7	6	3	4	3	3	3	2	2	2	2	2	1	2	1	1
2	LAURA	6	7	6	6	3	4	4	2	3	2	1	1	2	2	2	1	0	0
3	THERESA	7	6	8	6	4	4	4	3	4	3	2	2	3	3	2	2	1	1
4	BRENDA	6	6	6	7	4	4	4	2	3	2	1	1	2	2	2	1	0	0
5	CHARLOTTE	3	3	4	4	4	2	2	0	2	1	0	0	1	1	1	0	0	0
6	FRANCES	4	4	4	4	2	4	3	2	2	1	1	1	1	1	1	1	0	0
7	ELEANOR	3	4	4	4	2	3	4	2	3	2	1	1	2	2	2	1	0	0
8	PEARL	3	2	3	2	0	2	2	3	2	2	2	2	2	2	1	2	1	1
9	RUTH	3	3	4	3	2	2	3	2	4	3	2	2	3	2	2	2	1	1
10	VERNE	2	2	3	2	1	1	2	2	3	4	3	3	4	3	3	2	1	1
11	MYRNA	2	1	2	1	0	1	1	2	2	3	4	4	4	3	3	2	1	1
12	KATHERINE	2	1	2	1	0	1	1	2	2	3	4	6	6	5	3	2	1	1
13	SYLVIA	2	2	3	2	1	1	2	2	3	4	4	6	7	6	4	2	1	1
14	NORA	2	2	3	2	1	1	2	2	2	3	3	5	6	8	4	1	2	2
15	HELEN	1	2	2	2	1	1	2	1	2	3	3	3	4	4	5	1	1	1
16	DOROTHY	2	1	2	1	0	1	1	2	2	2	2	2	2	1	1	2	1	1
17	OLIVIA	1	0	1	0	0	0	0	1	1	1	1	1	1	2	1	1	2	2
18	FLORA	1	0	1	0	0	0	0	1	1	1	1	1	1	2	1	1	2	2

Figure 4.22. Southern Club Women: Co-Membership Network

attended at least one event with every other woman in the network, while Olivia and Flora attended none of the same events as Laura, Brenda, Charlotte, Frances, and Eleanor. The cell values along the diagonal have their own unique properties. They tell us how many total events each of the women attended. As you can see, Evelyn attended eight, Laura seven, Theresa eight, and Brenda seven, while Dorothy, Olivia, and Flora attended the fewest (two).

The event-overlap network (Figure 4.23) provides useful information as well. Here, the diagonal tells us how many women attended each event, while the off-diagonal cells tell us how many women each event "shared." In other words, three women attended events E1, E2, E13, and E14, while events E7 (ten women), E8 (fourteen women), and E9 (twelve women) were by far the most popular. And, events E1 and E2 shared two women (i.e., two women attended both E1 and E2), whereas events E8 and E9 shared nine women. In other words, by simply transforming a two-mode network into a one-mode network, we create helpful information about

		1 E1	2 E2	3 E3	4 E4	5 E5	6 E6	7 E7	8 E8	9 E9	10 E1	11 E1	12 E1	13 E1	14 E1
1	E1	3	2	3	2	3	3	2	3	1	0	0	0	0	0
2	E2	2	3	3	2	3	3	2	3	2	0	0	0	0	0
3	E3	3	3	6	4	6	5	4	5	2	0	0	0	0	0
4	E4	2	2	4	4	4	3	3	3	2	0	0	0	0	0
5	E5	3	3	6	4	8	6	6	7	3	0	0	0	0	0
6	E6	3	3	5	3	6	8	5	7	4	1	1	1	1	1
7	E7	2	2	4	3	6	5	10	8	5	3	2	4	2	2
8	E8	3	3	5	3	7	7	8	14	9	4	1	5	2	2
9	E9	1	2	2	2	3	4	5	9	12	4	3	5	3	3
10	E10	0	0	0	0	0	1	3	4	4	5	2	5	3	3
11	E11	0	0	0	0	0	1	2	1	3	2	4	2	1	1
12	E12	0	0	0	0	0	1	4	5	5	5	2	6	3	3
13	E13	0	0	0	0	0	1	2	2	3	3	1	3	3	3
14	E14	0	0	0	0	0	1	2	2	3	3	1	3	3	3

Figure 4.23. Southern Club Women: Event Overlap Network

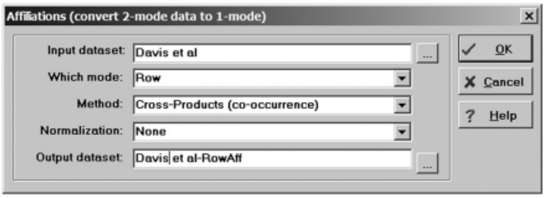


Figure 4.24. UCINET Affiliations Dialog Box

the network we are examining, long before we estimate more complicated metrics.

UCINET: One-Mode Networks from Two-Mode Networks. In UCINET you derive one-mode networks from two-mode networks (in this case, `Davis et al.##h`) using the *Data>Affiliations (2-mode to 1-mode)* command. This brings up a dialog box like the one illustrated in Figure 4.24. For an actor-by-actor matrix, choose "Row" in the "Which mode" drop-down menu since actors are generally listed in rows; for an event-by-event matrix, choose "Column" in the "Which mode" option since events (affiliations) are generally listed in columns.

[UCINET] Data >Affiliations (2-mode to 1-mode)

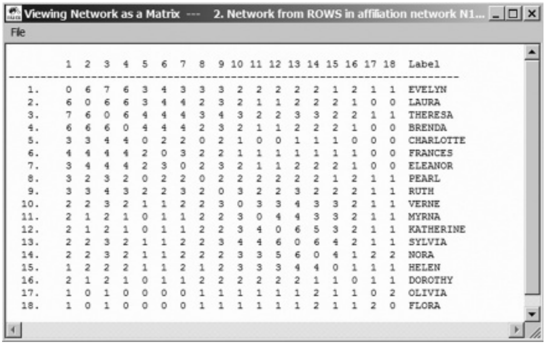
Be sure to save the files ("Output dataset") under different file names, otherwise if you derive both an actor-by-actor matrix and an event-by-event matrix, whichever one you derive last may overwrite any ones you derived earlier. Unfortunately, a warning box does not appear in UCINET when you are about to overwrite an already existing file. You can display the two newly created networks by either choosing the "Display" option found under the *Data* menu or by clicking on "D" icon located just below UCINET's menu bar. They should look similar to Figures 4.22 and 4.23.

Data>Display

Pajek: One-Mode Networks from Two-Mode Networks. Deriving one-mode networks from two-mode networks in Pajek is simple, but (like in UCINET) you need to know which actors/events are assigned to the rows and columns. First, we need to read the network data (`Davis et al.net`) into Pajek using the *File>Network>Read* command or click on the folder icon located under the "Networks" button (circled in Figure 3.10). To create an actor-by-actor (co-membership) matrix choose the "Rows" option under the *Net>Transform>2-Mode to 1-Mode* submenu (assuming that that actors appear are listed in rows). After issuing the command, the Report window will appear. Close this and you will see that a new network appears in the Network drop-down menu. Repeat the procedure, except choose the "Columns" option. If you want to display the co-membership network in matrix format, make sure that it is highlighted in the first or top Network drop-down menu, and then

[Pajek] File >Network >Read

Net >Transform >2-Mode to 1-Mode>Rows, Columns



Viewing Network as a Matrix --- 2. Network from ROWS in affiliation network N1...

File

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	Label
1.	0	6	7	6	3	4	3	3	3	2	2	2	2	2	1	2	1	1	EVELYN
2.	6	0	6	6	3	4	4	2	3	2	1	1	2	2	2	1	0	0	LAURA
3.	7	6	0	6	4	4	4	3	4	3	2	2	3	3	2	2	1	1	THERESA
4.	6	6	6	0	4	4	4	2	3	2	1	1	2	2	2	1	0	0	BRENDA
5.	3	3	4	4	0	2	2	0	2	1	0	0	1	1	1	0	0	0	CHARLOTTE
6.	4	4	4	4	2	0	3	2	2	1	1	1	1	1	1	1	0	0	FRANCES
7.	3	4	4	4	2	3	0	2	3	2	1	1	2	2	2	1	0	0	ELEANOR
8.	3	2	3	2	0	2	2	0	2	2	2	2	2	2	1	2	1	1	PEARL
9.	3	3	4	3	2	2	3	2	0	3	2	2	3	2	2	2	1	1	RUTH
10.	2	2	3	2	1	1	2	2	3	0	3	3	4	3	3	2	1	1	VERNE
11.	2	1	2	1	0	1	1	2	2	3	0	4	4	3	3	2	1	1	MYRNA
12.	2	1	2	1	0	1	1	2	2	3	4	0	6	5	3	2	1	1	KATHERINE
13.	2	2	3	2	1	1	2	2	3	4	4	6	0	6	4	2	1	1	SYLVIA
14.	2	2	3	2	1	1	2	2	2	3	3	5	6	0	4	1	2	2	NORA
15.	1	2	2	2	1	1	2	1	2	3	3	3	4	4	0	1	1	1	HELEN
16.	2	1	2	1	0	1	1	2	2	2	2	2	2	1	1	0	1	1	DOROTHY
17.	1	0	1	0	0	0	0	1	1	1	1	1	1	2	1	1	0	2	OLIVIA
18.	1	0	1	0	0	0	0	1	1	1	1	1	1	2	1	1	2	0	FLORA

Figure 4.25. Pajek Display of Davis's Southern Women Co-Membership Network

double-click anywhere on the Network drop-down menu. This will call up a dialog box (not shown) that asks what type of presentation you want: binary, valued, or lists. Generally, you will want to accept Pajek's default, which in this case is *valued*. Click "OK" and you should get an output similar to Figure 4.25.

Note that in Pajek, the diagonal cell values all equal zero. If we want Pajek to include diagonal values when we transform two-mode networks to one-mode networks, then prior to doing so, we need to first select Pajek's *Net>Transform>2-Mode to 1-Mode>Include Loops* option ("loops" is another term for diagonal cell values because they refer to ties that actors have to themselves). Be careful when doing this, however, because the inclusion of loops will alter centrality calculations in Pajek.

ORA: One-Mode Networks from Two-Mode Networks. To derive one-mode networks from two-mode networks in ORA, in the Meta-Network Manager panel highlight and right-click on the two-mode network (`Davis et al.xml`) you wish to transform. This brings up a set of options. From these, choose the "Fold" option (Figure 4.26), which is ORA's term for transforming a two-mode network into a one-mode network.

This brings up a dialog box (Figure 4.27), which asks the type of "folding" method you want to use, whether you want rows or columns, and the name that you want to call the new network. In terms of method, you will typically want to keep the "Shared Links" default. Here, we have chosen to look at the co-membership network (i.e., rows) and accepted ORA's default name for the new file. Clicking on the "Fold" button creates a new network in the meta-network that you can examine under the Editor tab or visualize with ORA's visualizer.

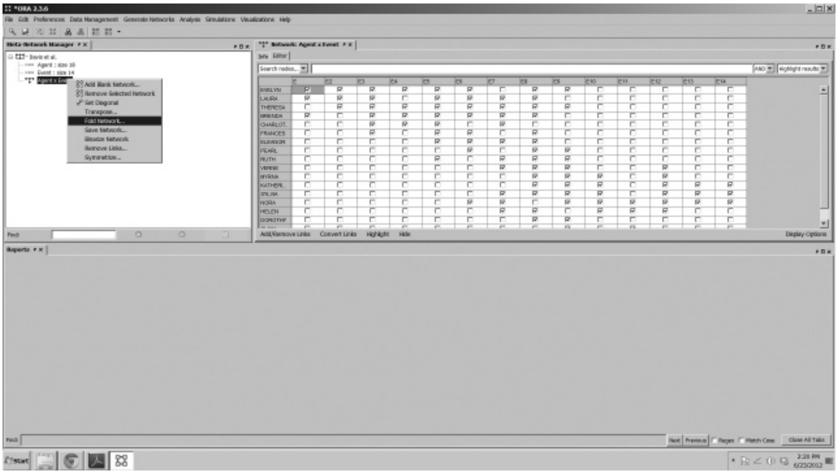


Figure 4.26. Transforming Two-Mode Network into One-Mode Network In ORA

4.8 Combining, Aggregating, and Parsing Networks

Up to this point we have focused on single types of relationships among actors: friendship, advice, attendance at a particular event, and so on. In the real world, however, actors are typically involved in more than one type of relation (Hanneman and Riddle 2005). As we have already seen, most individuals are embedded in several types of ties (e.g., friendship, kinship, and economic), and corporate and state actors are no different. Businesses engage in financial and informational exchanges and sometimes form alliances with one another (Saxenian 1994), while countries are linked through numerous cultural, economic, military, and political ties, not to mention transnational corporations, nongovernmental

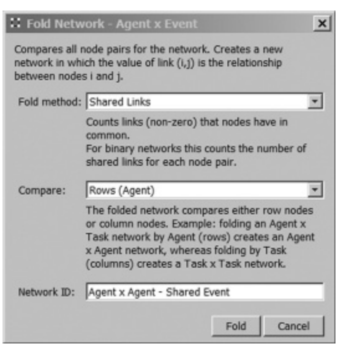


Figure 4.27. ORA's Fold Network Dialog Box

organizations, and international agencies (Meyer et al. 1997). More important, different types of ties can pressure actors to make conflicting choices (Simmel [1908, 1922] 1955). Thus, being able to combine, aggregate, and parse multiple relations is highly important and something we explore in this section. We first examine UCINET because its tools for manipulating network data are excellent. We then consider NetDraw's capabilities for visualizing multiple networks that work quite nicely with UCINET datasets. Next, comes Pajek, which is not as user-friendly as UCINET when it comes to manipulating data but has powerful visualization tools. Finally, we consider ORA, which provides a number of helpful functions for manipulating and visualizing networks.

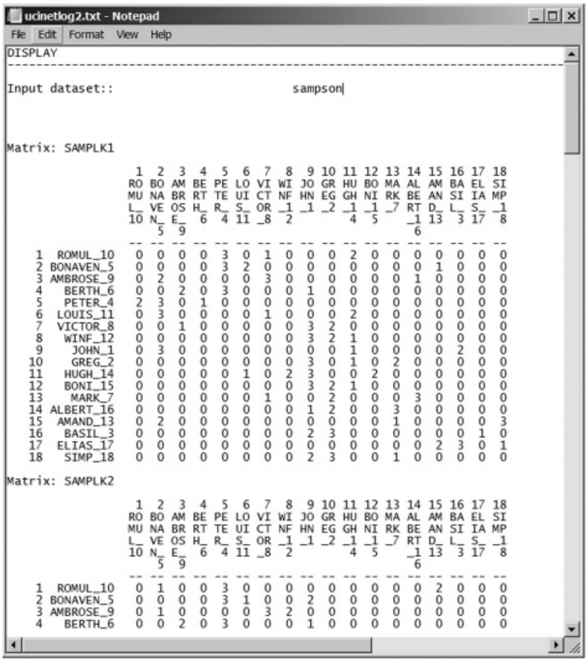
In this section we also introduce the Noordin Top Terrorist Network data, combining, aggregating, and parsing it into the networks we will explore throughout the book. The data are discussed in detail in Appendix 1. Before turning our attention to these data, however, we will first examine a somewhat simpler network dataset: the Sampson Monastery dataset, which was recorded by Samuel Sampson (1968) who, for his Ph.D. dissertation, spent a year in a Roman Catholic monastery in the late 1960s observing the social interactions among a group of monks. During his stay, a "crisis in the cloister" developed in reaction to the changes introduced by Vatican II that resulted in the expulsion of four monks and the voluntary departure of several others.[7] In the end, only four monks remained (Bonaventure, Berthold, Ambrose, and Louis). While he was there, Sampson coded four types of relational data that he further subdivided into positive and negative ties. He had each monk rank his top three choices for each type of relation, although some offered tied ranks for their top four choices. The relations he recorded were esteem (SAMPES) and disesteem (SAMPDES), liking (SAMPLK – three different time periods were recorded) and disliking (SAMPDLK – only one time period), positive (SAMPIN) and negative influence (SAMPNIN), and praise (SAMPPR) and blame (SAMPNPR). In each network, a "3" indicates the monk's highest or first choice and a "1" indicates his last choice.

Multirelational Data in UCINET and NetDraw

UCINET includes a variety of tools for analyzing multiplex data. Some allow you to extract an individual network from a multiple network file, while others allow you to combine separate data files. One of the most common ways of storing multiple network data is by "stacking" a set of actor-by-actor networks, one for each type of relation. Figure 4.28 displays part of the output from a *Data>Display* command for the Sampson

[UCINEL] Data>Display

[7] Vatican II was a conference of all the bishops and cardinals of the Roman Catholic Church who met from 1962 to 1965 in which numerous changes were introduced in order to modernize the Church (Finke and Stark 2005; Stark and Finke 2000).



ucinetlog2.txt - Notepad

File Edit Format View Help

DISPLAY

Input dataset:: sampson

Matrix: SAMPLK1

		1 ROMUL_10	2 BONAVEN_5	3 AMBROSE_9	4 BERTH_6	5 PETER_4	6 LOUIS_11	7 VICTOR_8	8 WINF_12	9 JOHN_1	10 GREG_2	11 HUGH_14	12 BONI_15	13 MARK_7	14 ALBERT_16	15 AMAND_13	16 BASIL_3	17 ELIAS_17	18 SIMP_18
1	ROMUL_10	0	0	0	0	3	0	1	0	0	0	2	0	0	0	0	0	0	0
2	BONAVEN_5	0	0	0	0	3	2	0	0	0	0	0	0	0	0	1	0	0	0
3	AMBROSE_9	0	2	0	0	0	0	3	0	0	0	0	0	0	1	0	0	0	0
4	BERTH_6	0	0	2	0	3	0	0	0	1	0	0	0	0	0	0	0	0	0
5	PETER_4	2	3	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	LOUIS_11	0	3	0	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0
7	VICTOR_8	0	0	1	0	0	0	0	0	3	2	0	0	0	0	0	0	0	0
8	WINF_12	0	0	0	0	0	0	0	0	3	2	1	0	0	0	0	0	0	0
9	JOHN_1	0	3	0	0	0	0	0	0	0	0	1	0	0	0	0	2	0	0
10	GREG_2	0	0	0	0	0	0	0	0	3	0	1	0	2	0	0	0	0	0
11	HUGH_14	0	0	0	0	0	1	0	2	3	0	0	2	0	0	0	0	0	0
12	BONI_15	0	0	0	0	0	0	0	0	3	2	1	0	0	0	0	0	0	0
13	MARK_7	0	0	0	0	0	0	1	0	0	2	0	0	0	3	0	0	0	0
14	ALBERT_16	0	0	0	0	0	0	0	0	1	2	0	0	3	0	0	0	0	0
15	AMAND_13	0	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	3
16	BASIL_3	0	0	0	0	0	0	0	0	2	3	0	0	0	0	0	0	1	0
17	ELIAS_17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	3	0	1
18	SIMP_18	0	0	0	0	0	0	0	0	2	3	0	0	1	0	0	0	0	0

Matrix: SAMPLK2

		1 ROMUL_10	2 BONAVEN_5	3 AMBROSE_9	4 BERTH_6	5 PETER_4	6 LOUIS_11	7 VICTOR_8	8 WINF_12	9 JOHN_1	10 GREG_2	11 HUGH_14	12 BONI_15	13 MARK_7	14 ALBERT_16	15 AMAND_13	16 BASIL_3	17 ELIAS_17	18 SIMP_18
1	ROMUL_10	0	1	0	0	3	0	0	0	0	0	0	0	0	0	2	0	0	0
2	BONAVEN_5	0	0	0	0	3	1	0	0	2	0	0	0	0	0	0	0	0	0
3	AMBROSE_9	0	1	0	0	0	0	3	2	0	0	0	0	0	0	0	0	0	0
4	BERTH_6	0	0	2	0	3	0	0	0	1	0	0	0	0	0	0	0	0	0

Figure 4.28. Sampson Monastery Data (UCINET Log)

Monastery dataset (`sampson.##h`). At the top of the log/output, you can see the entire "Liking, Time One" dataset, and just below you can see a portion of the "Liking, Time Two" dataset. Scrolling down through UCINET's output displays the other networks that are included in the dataset.

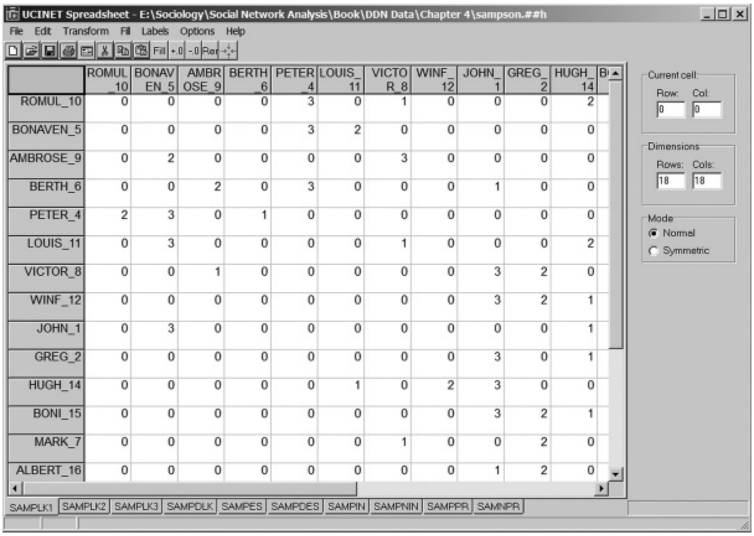
You can also examine stacked network datasets using UCINET's spreadsheet function, which if you recall, is accessed with UCINET's *Data>Data Editors>Matrix Editor* command. Once this is called up, you can open any UCINET file using the *File>Open* command, which brings up a dialog box (not shown) that enables you to identify the network data you wish to open. The Sampson data are displayed in Figure 4.29. As you can see, UCINET's spreadsheet function is similar to commercial spreadsheet programs. Each type of relation appears under a different tab that you can click on and examine. In this case, the "Liking, Time One" data are displayed.

Data >Data Editors >Matrix Editor

[Spread Sheet] File>open

Next, let us examine at how to unpack stacked network data in UCINET. To do this, select the *Data>Unpack* command. This brings up an "Unpack" dialog box (Figure 4.30) that asks you for the input dataset and which relations to unpack. You can choose to unpack all of the relations or just some. UCINET's default is "All," but if you wanted to only unpack some, click on the "L" radio button, which brings up an additional dialog box that lets you pick the matrices of your choice. In this

Data>Unpack



	ROMUL_10	BONAVEN_5	AMBROSE_9	BERTH_6	PETER_4	LOUIS_11	VICTOR_8	WINF_12	JOHN_1	GREG_2	HUGH_14
ROMUL_10	0	0	0	0	3	0	1	0	0	0	2
BONAVEN_5	0	0	0	0	3	2	0	0	0	0	0
AMBROSE_9	0	2	0	0	0	0	3	0	0	0	0
BERTH_6	0	0	2	0	3	0	0	0	1	0	0
PETER_4	2	3	0	1	0	0	0	0	0	0	0
LOUIS_11	0	3	0	0	0	0	1	0	0	0	2
VICTOR_8	0	0	1	0	0	0	0	0	3	2	0
WINF_12	0	0	0	0	0	0	0	0	3	2	1
JOHN_1	0	3	0	0	0	0	0	0	0	0	1
GREG_2	0	0	0	0	0	0	0	0	3	0	1
HUGH_14	0	0	0	0	0	1	0	2	3	0	0
BONI_15	0	0	0	0	0	0	0	0	3	2	1
MARK_7	0	0	0	0	0	0	1	0	0	2	0
ALBERT_16	0	0	0	0	0	0	0	0	1	2	0

Figure 4.29. Sampson Monastery Data (UCINET Spreadsheet Editor)

case, we have kept UCINET's default, which means that clicking "OK" will unpack ten separate matrices/networks. You can examine each of the new networks using either UCINET's display or spreadsheet functions. UCINET also allows you to join separate datasets into a single stacked dataset; we will see how to do this when we examine the Noordin Top network data.

Now, let's see how we can visualize the Sampson data using NetDraw, which allows users to view stacked networks with different colored lines. Open the `sampson.##h` dataset, using the *File>Open>Ucinet dataset>Network* command. Recall that the Rels tab (see Figure 4.31) allows you to select which network to view, which can be useful for combining and switching back and forth between relations. In this case, four relations have been selected. You can assign different colors to the various relations by using the dialog box (not shown) that the *Properties>Lines>Color>Relation* command calls up. Here, because we cannot display colors, the relations appear in gray scale, but typically it is easiest to accept NetDraw's default colors. Note that when two actors share more than one relation (e.g., liking and esteem), NetDraw assigns a

[NetDraw] File>Open >Ucinet dataset >Network

Properties >Lines >Color >Relation

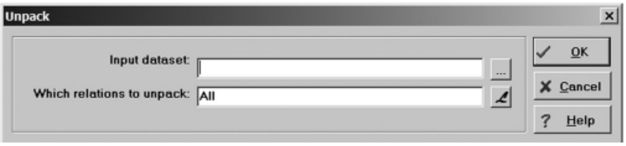


Figure 4.30. UCINET Unpack Dialog Box

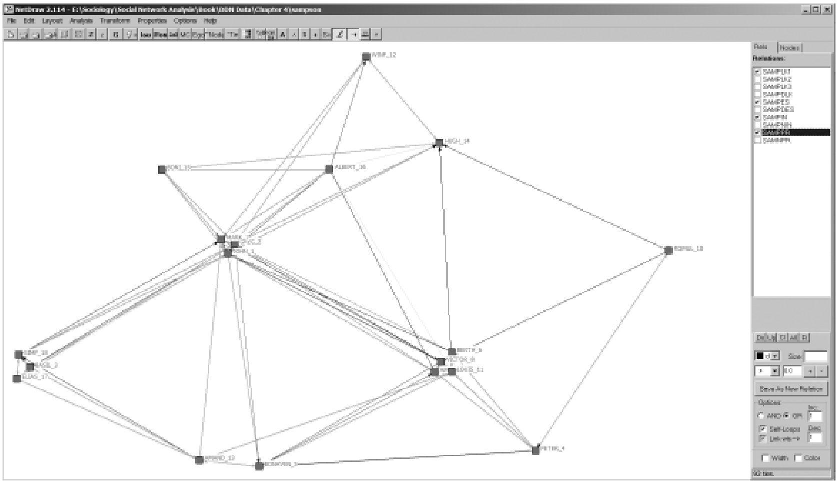


Figure 4.31. NetDraw Drawing of Sampson's Monastery Data

separate color to that tie. In Figure 4.31 multiple relations ties are black, but NetDraw's default is usually gray.

Combining Networks in UCINET. It is easy in UCINET to combine separate networks into a stacked dataset, which we will illustrate using the Noordin Top Terrorist Network data, which were drawn primarily from *Terrorism in Indonesia: Noordin's Networks*, a publication of the International Crisis Group (2006; see Appendix 1). We will begin by aggregating and analyzing the following relations into four types of "networks," which are adapted from a typology suggested by Krebs (2001:51) and summarized in Table 4.2.[8]

The classmate, friendship, kinship, soulmate, and internal communication networks were originally coded as one-mode networks, while the meeting, operations, training, and business and finance networks were derived from two-mode networks. The logistics network was parsed from two two-mode networks – logistical place and logistical function – that

8 As noted in the preface, the Noordin network serves as a running example throughout this book. In subsequent chapters these trust, operational, communication, and business and finance networks will sometimes be combined into a single network. Moreover, subnetworks of these four networks (five if you count the combined network) will be extracted based on whether particular actors are dead, alive, or in jail (see Appendix 1). In other words, the four original networks plus the combined network can each be examined as a whole, as well as parsed into and analyzed as separate subnetworks. We will not analyze each of the twenty possible variations on Noordin's network in every chapter because to do so would be needlessly repetitive. Instead, in each chapter, networks are chosen that help illustrate the algorithm under consideration. When analyzing dark networks in the real world, analysts will want to consider (within reason) all possible permutations of a network before they begin to craft strategies for its disruption.

Table 4.2. *Aggregation of Noordin Top's terrorist network ties*

	Network			
	Trust	Operational	Communications	Finance
Relations	• Classmates • Friendship • Kinship • Soulmates	• Logistics • Meetings • Operations • Training	• Communications	• Business and Finance

were converted to two one-mode networks. In order for there to be a tie between two individuals in the logistics network, they had to share a tie in both the logistical place and logistical function networks.[9] Of course, one could argue that this sorting of relations could have been done differently. Perhaps ties made at school (i.e., classmates) should not be considered ties of trust, or that business and finance ties should be included in the operational network. But that would be missing the point. As we saw in the second chapter, which networks are chosen to be combined, aggregated, or parsed is part of the process for crafting strategies. The important thing is to be clear with what you do so that others may follow (and sometimes disagree) with your analysis.

[UCINET] Data>Join

Figure 4.32 illustrates the use of the *Data>Join* command for creating the Noordin trust network. As you can see the classmates (`Classmates.##h`), friendship (`Friendship.##h`), kinship (`Kinship.##h`), and soulmates (`Soulmates.##h`) networks have been selected for aggregation, and that the name of the new network is "Trust Network" (UCINET's default name is "Join"). Note also that the "Matrices" option under the Dimensions to Join panel has been selected because we are combining matrices. You can also combine the rows of two or more matrices (keeping the columns the same) or the columns of two or more matrices (keeping the rows the same). This latter option is useful for combining and analyzing series of two-mode networks (e.g., schools and faith communities).

Transform >Matrix Operations >Within dataset > Aggregations

Aggregating Networks in UCINET. Sometimes, you may want to create a single-valued network from a series of stacked networks. To do this we need to use the *Transform>Matrix Operations>Within dataset>Aggregations* command, which calls up a dialog box similar to Figure 4.33. In this case, the operational network has been selected, which if you recall, is a stacked network consisting of the logistics

[9] How to parse two (or more) networks in such a way is discussed further on in the chapter.

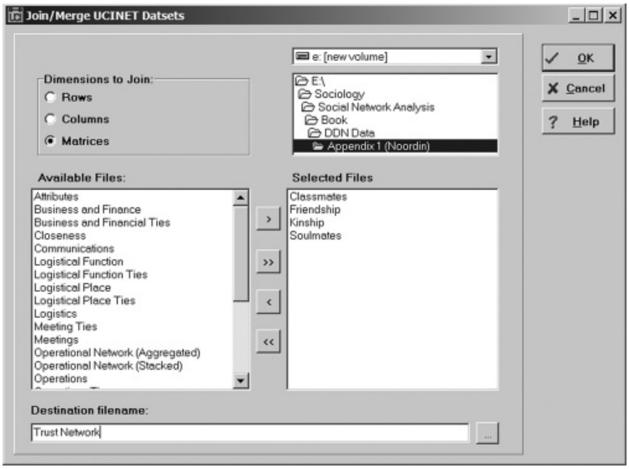


Figure 4.32. UCINET's Join/Merge Datasets Function

(`Logistics.##h`), meetings (`Meetings.##h`), operations (`Operations.##h`), and training (`Training.##h`) networks (see Table 4.2). This function provides several different arithmetic operation options – you can sum, average, and get the maximum, minimum, or standard deviation of the corresponding cells in each network. Here, we are asking UCINET to sum the networks, which means that UCINET will aggregate the number of times each pair of actors share a tie. In other words, if two actors share logistic and operations ties, then the value of their tie in the aggregated network will equal two. If they share logistic, operations, and training ties, then the value of their tie will equal three.

You can also aggregate networks of the same size that have not been combined into a stacked network using the *Transform>Matrix Operations>Between datasets>Statistical Summaries* command, which brings up a dialog box (Figure 4.34). Here, the networks that were previously combined into the operational network have been selected. Clicking "OK" combines the matrices into a single-valued matrix that should be identical to the one we created previously using the within dataset aggregations command.

Transform >Matrix Operations >Between datasets >Statistical Summaries

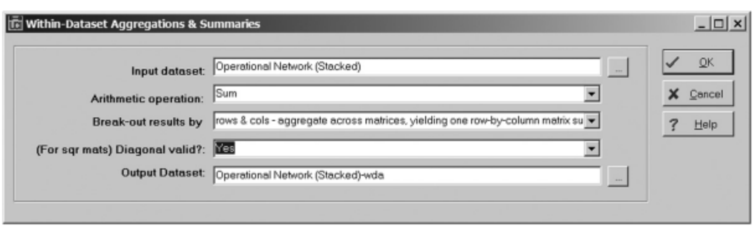


Figure 4.33. UCINET within Dataset Aggregations Dialog Box

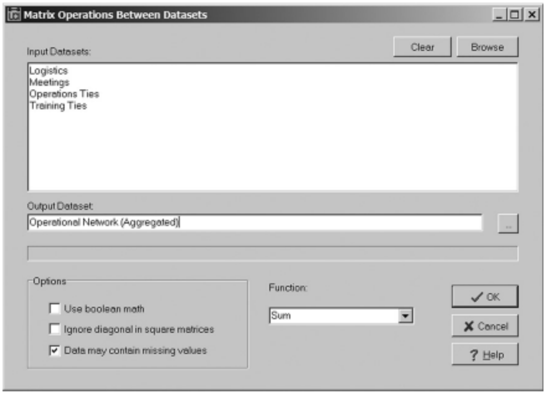


Figure 4.34. UCINET between Dataset Statistical Summaries Dialog Box

Parsing Networks in UCINET. Just because two people are members of the same organization or attend the same event does not mean that they necessarily are friends or even know each other. Thus, we may want to use some sort of threshold before concluding that a tie actually exists between two actors. Taking the Noordin logistical place and logistical function networks as an example, let us assume that a tie only exists between two actors if they shared *both* the same logistical location and logistical function. For example, if they are both located in Ambon and handle transportation functions, then we will assume a tie exists between the two actors. However, if they are both located in Ambon but do not share the same functional responsibilities, or if they share the same functional responsibilities but are located in different cities, then we will assume that a tie does not exist between them.

Transform >Matrix Operations >Between datasets >Boolean Combinations

To do this, we take the co-membership networks of the logistical place (`Logistical Place.##h`) and logistical function (`Logistical Function.##h`) two-mode networks and use UCINET's *Transform>Matrix Operations>Between datasets>Boolean Combinations* command. The resulting dialog box (Figure 4.35) is nothing more than a means for testing if/then statements. As you can see, we have

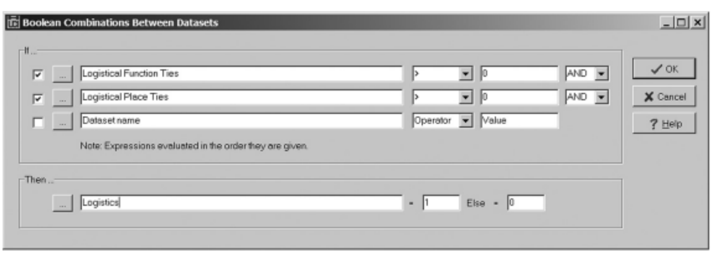
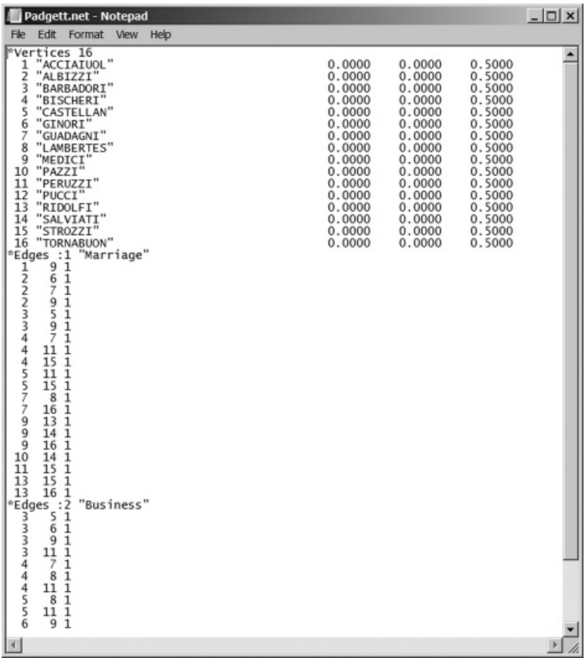


Figure 4.35. UCINET Boolean Combination Dialog Box



Padgett.net - Notepad

File Edit Format View Help

```
*Vertices 16
 1 "ACCIAIUOL"                                0.0000    0.0000    0.5000
 2 "ALBIZZI"                                  0.0000    0.0000    0.5000
 3 "BARBADORI"                                0.0000    0.0000    0.5000
 4 "BISCHERI"                                 0.0000    0.0000    0.5000
 5 "CASTELLAN"                                0.0000    0.0000    0.5000
 6 "GINORI"                                   0.0000    0.0000    0.5000
 7 "GUADAGNI"                                 0.0000    0.0000    0.5000
 8 "LAMBERTES"                                0.0000    0.0000    0.5000
 9 "MEDICI"                                   0.0000    0.0000    0.5000
10 "PAZZI"                                    0.0000    0.0000    0.5000
11 "PERUZZI"                                  0.0000    0.0000    0.5000
12 "PUCCI"                                    0.0000    0.0000    0.5000
13 "RIDOLFI"                                  0.0000    0.0000    0.5000
14 "SALVIATI"                                 0.0000    0.0000    0.5000
15 "STROZZI"                                  0.0000    0.0000    0.5000
16 "TORNABUON"                                0.0000    0.0000    0.5000
*Edges :1 "Marriage"
 1   9 1
 2   6 1
 2   7 1
 2   9 1
 3   5 1
 3   9 1
 4   7 1
 4  11 1
 4  15 1
 5  11 1
 5  15 1
 7   8 1
 7  16 1
 9  13 1
 9  14 1
 9  16 1
10  14 1
11  15 1
13  15 1
13  16 1
*Edges :2 "Business"
 3   5 1
 3   6 1
 3   9 1
 3  11 1
 4   7 1
 4   8 1
 4  11 1
 5   8 1
 5  11 1
 6   9 1
```

Figure 4.36. Pajek Edgelist of Padgett Marriage and Business Network Data

selected the one-mode network derived from the logistical function network in the first network file box and the one-mode network derived from the logistical place network in the second. To the right of both of these boxes, we have chosen the ">" sign in both operator boxes and a "0" in the value boxes; these tell UCINET to look for every instance where the tie between each pair of actors is greater than zero. When this is true, UCINET will assign a value of one for each tie because in the last (bottom) network file box we have indicated the value for each tie, which if both statements are true, should be one. In this final box, we have also assigned a name for the new file that will be created (i.e., "Logistics") when we click "OK."

Multirelational Data in Pajek

How does Pajek store multirelational network data? It does so by assigning relation numbers to a set of ties (i.e., arcs, edges), as Figure 4.36 illustrates with the Padgett marriage and business networks (`Padgett.net`). The figure shows that, after a list of the vertices (i.e., actors) in the network, Pajek includes a separate edge (or arc) list for each relation, each of which is preceded by a number and name.

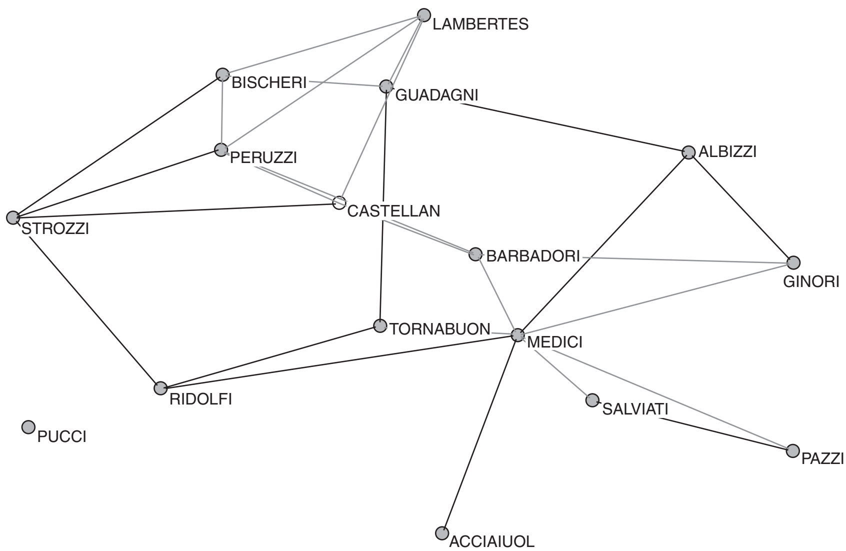


Figure 4.37. Pajek Network Map of Padgett Multirelational Data

[Pajek – Draw Screen] Data >Export >DL

Data >Export >Pajek

Looking at Figure 4.36 you can see that the list of the sixteen families in the network is followed first by the marriage edge list then the business edge list (only a portion of the latter edge list appears in the figure).[10] By and large you will use multirelational data that have been recorded in and exported from UCINET using the latter's *Data>Export>DL* command (rather than the *Data>Export>Pajek* command), which is the same command we used earlier to export two-mode networks into Pajek-readable formats.

Options >Colors>Edges >Relation Number

Options >Colors>Arcs >Relation Number

Options>Colors >Relation Colors

Options >Lines >Draw Lines >Relations Ins

As with NetDraw in Pajek we can represent the relation number of a line-by-line color; this is accomplished in Pajek's Draw screen with the *Options>Colors>Edges>Relation Number* (for edges) and *Options>Colors>Arcs>Relation Number* (for arcs) commands. We can choose the color of each relation number in the *Options>Colors>Relation Colors* dialog screen. Figure 4.37 presents a network map of the Padgett data where marriage ties are colored black and the business ties are colored gray. When two actors share a tie (e.g., Bischeri and Peruzzi), Pajek colors the ties by the last relation in the file (in this case, business = gray). This differs from NetDraw, which you will recall colors multiple relations gray. We can also choose to display some but not all of the relations with the *Options>Lines>Draw Lines>Relations Ins*

[10] The first line of the marriage edge list indicates that vertex 1 (the Acciaiuols) shares a tie with vertex 9 (the Medicis) of strength one; the first line of the business edge list indicates that vertex 3 (the Barbadoris) shares a tie with vertex 5 (the Castellans).

Figure 4.38. Two Networks Highlighted in Pajek's Network Drop-Down Menus

command, which calls up a dialog box where you indicate which relations (numbers) you want to view.

Pajek also allows users to extract one or more relations from a multiple relations network with its *Net>Transform>Multiple Relations>Extract Relation(s)* command (accessed at Pajek's main screen). When we issue this command, Pajek generates a new network for each of the selected relation numbers, preserving the relation number and name. We can also recode relation number and change relation labels with the *Net>Transform>Multiple Relations>Change Relation Number-Label* command. We can also change the label name of a relation by opening and editing the Pajek file in a text editor such as Notepad. That is how the labels were added to Figure 4.37.

[Main Screen] Net>Transform >Multiple Relations >Extract Relation(s)

Net>Transform >Multiple Relations >Change Relation Number-Label

Aggregating Networks in Pajek. In Pajek we can only combine two networks at a time, which is why when you are working with more than two, you will probably want to combine and aggregate in UCINET. Nevertheless, we briefly illustrate how to aggregate files in Pajek using the Padgett network data. First open the Padgett project file (`Padgett.paj`) using Pajek's *File>Pajek Project File>Read* command (or you can use the "F1" button). Next, click on the "Network" radio button (circled in Figure 4.38) in order to display two Network drop-down menus (Figure 4.38). In the first, display either the marriage or business network, and in the second, display the other. Then, select the *Nets>Union of Lines* command. This will create a new network that Pajek labels *Fusion of 2 and 3*, which you can rename using the *File>Network>Change Label* command. When Pajek combines networks using the *Union of Lines* command, it treats them as a single relation with multiple lines. If you want to combine the two networks into a single-valued network, make sure that the newly created (i.e., fused) network is highlighted in the first network drop list, then select the *Sum Values* command under the *Net>Transform>Remove>Multiple lines* submenu.

Pajek's File>Pajek Project File>Read

Nets>Union of Lines

File>Network >Change Label

Net>Transform >Remove >Multiple lines

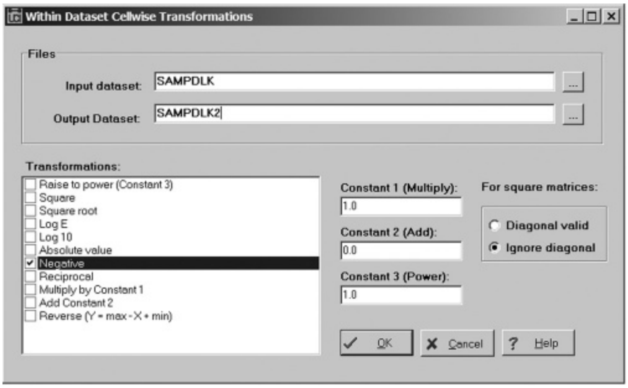


Figure 4.39. UCINET within Dataset Cellwise Transformations Dialog Box

Excursus: Positive and Negative Relations in UCINET and Pajek

Before examining how to work with multirelational data in ORA, we will first explore one way of simultaneously working with positive and negative ties. Here we will use the Sampson network data because it contains both positive (e.g., like) and negative (e.g., dislike) ties; however, the techniques that are illustrated can be applied to any dark network data that contain positive (e.g., friend) and negative (e.g., enemy) ties. In this section, we will multiply one of the negative networks ("dislike") by "–1" in order to transform the values into "negative" values and then add this transformed network to a positive network ("like" at time 3), which will give us a network that contains both positive and negative ties. We will then export this network to Pajek in order to visualize it.

[UCINET] Transform> Matrix Operations> Within dataset>Cellwise Transformations

UCINET. Let's begin by transforming these positive values into negative ones using the *Transform>Matrix Operations>Within dataset>Cellwise Transformations* command. In the resulting dialog box (Figure 4.39) note that we selected `SAMPDLK.##h` as our input dataset, named the transformed dataset `SAMPDLK2.##h`, and have checked the "Negative" box among the "Transformation" options. Click "OK" and UCINET should produce an output log (not shown) that indicates that the values in the new network are negative.

Transform> Matrix Operations> Between datasets> Statistical Summaries

The next step is to add this newly created network to the `SAMPLK3.##h` network with UCINET's *Transform>Matrix Operations>Between datasets>Statistical Summaries* command. This brings up a dialog box (Figure 4.40) where we use the "Browse" button to locate the files we intend to aggregate (`SAMPDLK2.##h` and `SAMPLK3.##h`) and assign a name to the output file (`SAMPLKDLK.##h`).

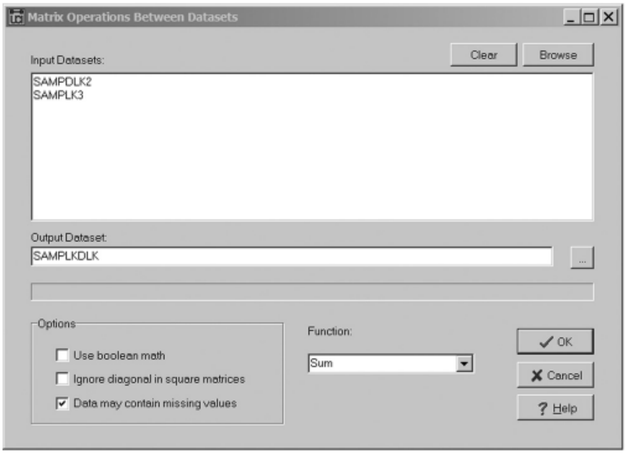


Figure 4.40. UCINET between Dataset Statistical Summaries Dialog Box

After clicking "OK," UCINET generates a new file and produces an output log that shows an aggregated network containing both positive and negative numbers (Figure 4.41).

The next step is to export this newly created aggregated network using the *Data>Export>DL* command (not shown) so that it is readable in Pajek. As before, we need to change the extension of the exported file to *.dat. The reason for using this command to export the data rather than the *Data>Export>Pajek>Network* command is because (at least currently) the latter command exports negative ties as positive ties.

Data>Export >DL

Data>Export >Pajek>Network

Pajek. Open Pajek and read in the network data you just exported from UCINET using the *File>Network>Read* command (remember that the network file you are trying to read has a *.dat extension not a *.net. one). Next, visualize the network using the *Draw>Draw* command and energize it using either the Kamada-Kawai or Fruchterman Reingold layout algorithms (or both; Figure 4.41). Note that there are both solid lines and negative lines between the nodes. In Pajek, solid lines indicate positive connections while dotted lines indicate negative connections. Moreover, as long as the *Options>Values of Lines>Similarities* option has been selected, the positive lines will pull nodes closer together while the negative lines will push them farther apart. This should, in theory, provide us with a better picture of the social closeness and distance between the monks.

[Pajek – Main Screen] File>Network >Read

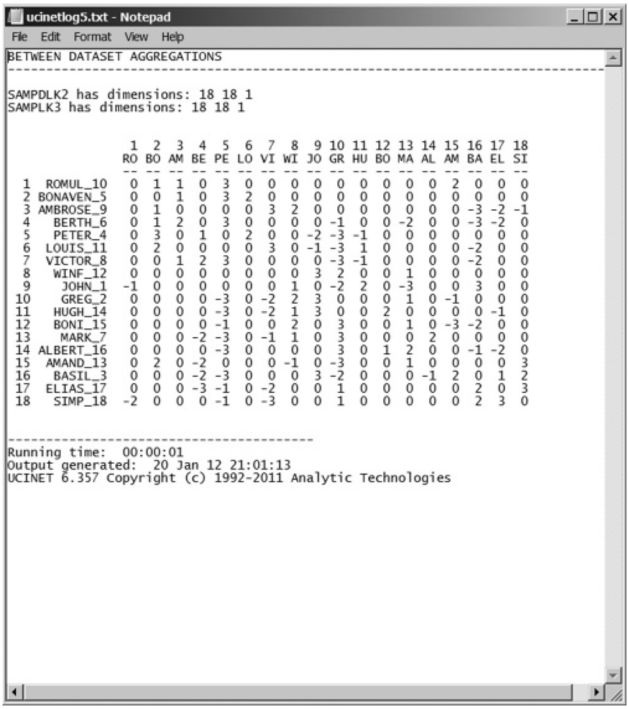
Draw>Draw

[Draw Screen] Layout>Energy >Kamada-Kawai >Free

Options >Values of Lines >Similarities

If the nodes are hard to distinguish with all of the lines, you can increase the size of the node using the *Options>Size>of Vertices* command in Pajek's Draw screen. Figure 4.41 uses a vertex size of "10." Generally,

Options >Size>of Vertices



ucinetlog5.txt - Notepad

File Edit Format View Help

```
BETWEEN DATASET AGGREGATIONS
--------------------------------------------------------------------------------

SAMPDLK2 has dimensions: 18 18 1
SAMPLK3 has dimensions: 18 18 1

                  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18
                 RO BO AM BE PE LO VI WI JO GR HU BO MA AL AM BA EL SI
                 -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- -- --
 1   ROMUL_10     0  1  1  0  3  0  0  0  0  0  0  0  0  0  2  0  0  0
 2 BONAVEN_5      0  0  1  0  3  2  0  0  0  0  0  0  0  0  0  0  0  0
 3 AMBROSE_9      0  1  0  0  0  0  3  2  0  0  0  0  0  0  0 -3 -2 -1
 4    BERTH_6     0  1  2  0  3  0  0  0  0 -1  0  0 -2  0  0 -3 -2  0
 5    PETER_4     0  3  0  1  0  2  0  0 -2 -3 -1  0  0  0  0  0  0  0
 6   LOUIS_11     0  2  0  0  0  0  3  0 -1 -3  1  0  0  0  0 -2  0  0
 7   VICTOR_8     0  0  1  2  3  0  0  0  0 -3 -1  0  0  0  0 -2  0  0
 8    WINF_12     0  0  0  0  0  0  0  0  3  2  0  0  1  0  0  0  0  0
 9     JOHN_1    -1  0  0  0  0  0  0  1  0 -2  2  0 -3  0  0  3  0  0
10     GREG_2     0  0  0  0 -3  0 -2  2  3  0  0  0  1  0 -1  0  0  0
11    HUGH_14     0  0  0  0 -3  0 -2  1  3  0  0  2  0  0  0  0 -1  0
12    BONI_15     0  0  0  0 -1  0  0  2  0  3  0  0  1  0 -3 -2  0  0
13     MARK_7     0  0  0 -2 -3  0 -1  1  0  3  0  0  0  2  0  0  0  0
14  ALBERT_16     0  0  0  0 -3  0  0  0  0  3  0  1  2  0  0 -1 -2  0
15   AMAND_13     0  2  0 -2  0  0  0 -1  0 -3  0  0  1  0  0  0  0  3
16    BASIL_3     0  0  0 -2 -3  0  0  0  3 -2  0  0  0 -1  2  0  1  2
17   ELIAS_17     0  0  0 -3 -1  0 -2  0  0  1  0  0  0  0  0  2  0  3
18    SIMP_18    -2  0  0  0 -1  0 -3  0  0  1  0  0  0  0  0  2  3  0


----------------------------------------
Running time:  00:00:01
Output generated:  20 Jan 12 21:01:13
UCINET 6.357 Copyright (c) 1992-2011 Analytic Technologies
```

Figure 4.41. UCINET between Dataset Aggregations Output Log

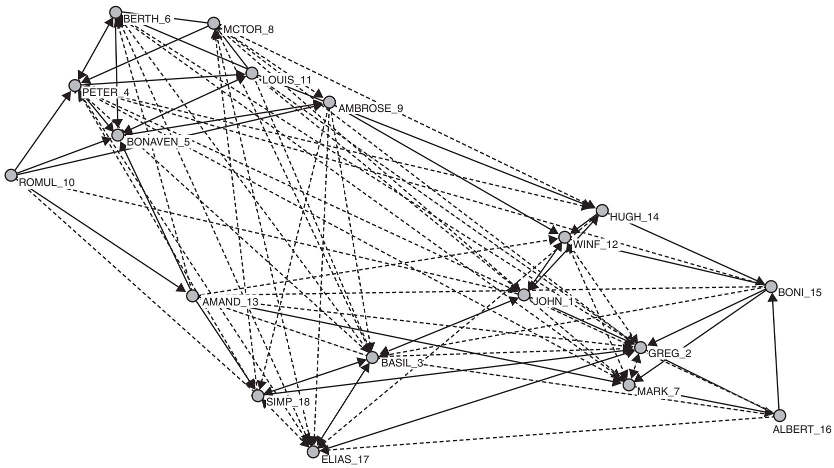


Figure 4.42. Pajek Drawing of Sampson Monastery Liking and Disliking Data

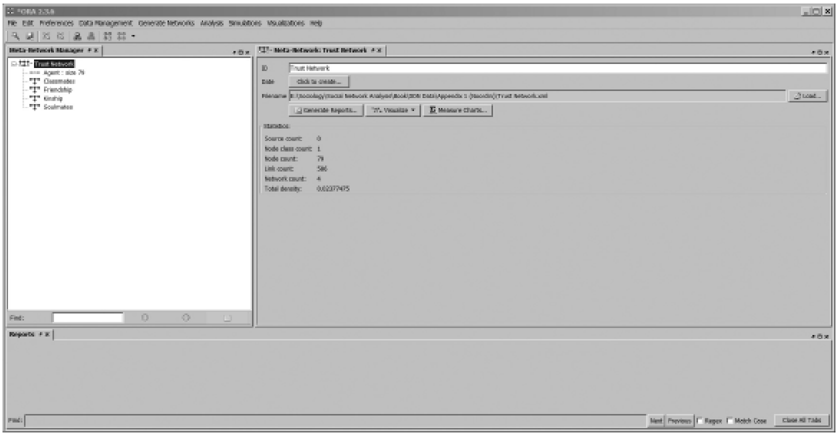


Figure 4.43. Noordin's Trust Network Loaded in ORA

however, it is wise to use the default setting ("0"), which tells Pajek to automatically set the size of the vertices. This is an especially useful option when node size reflects an attribute that varies considerably (e.g., age, centrality, etc.).

Multirelational Data in ORA

We have already seen in Section 4.5 how ORA stores relational data using its meta-network approach (see Figure 4.20). Network data can be entered directly using ORA's editor function (see Figures 4.9 and 4.10) or imported with its data import wizard (see Figures 4.18 and 4.19). One of ORA's more helpful features is that it imports network data that have been "stacked" in UCINET as a multirelational network. For example, using *UCINET's File>Data Import Wizard*, Noordin's multirelational trust network (`Trust Network (Stacked).##h`) was imported into ORA (Figure 4.43). As we will see later, ORA's analysis reports allow users to acquire metrics on one or all of the networks included in a meta-network.

[ORA] File>Data Import Wizard

To visualize this network, simply click on the "Visualize" speed button located in the Network Information/Editor panel or use the *Visualizations>View Networks>2D Visualization* command (Figure 4.44). As we discussed earlier, you can visualize all of the different types of relations at the same time, or (as we do in NetDraw) check the boxes of the relations you want to visualize or uncheck the boxes of the relations you do not. You can also alter the color of the ties by clicking on the name of the relation in the Legend box, which brings up an edge color dialog box (not shown) that allows you to choose an edge color for each type of tie. In Figure 4.43 each of the ties is colored a different scale of

Visualizations >View Networks>2D Visualization

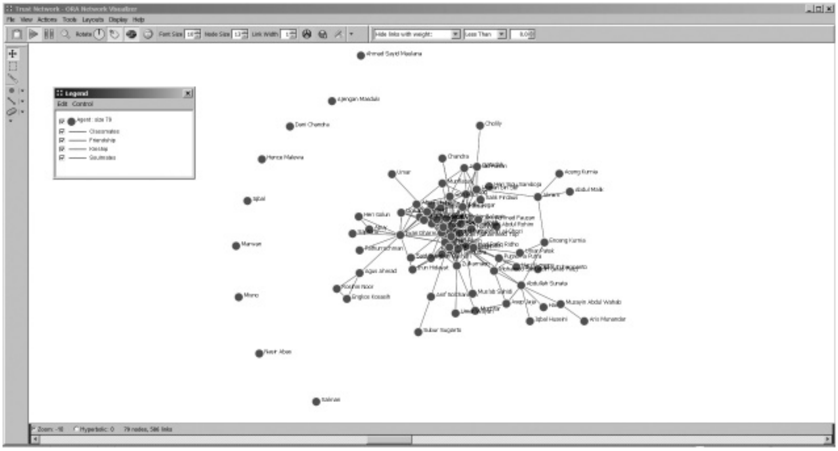


Figure 4.44. ORA Visualization of Noordin's Trust Network

gray. For most purposes, however (e.g., presentations), you will want to choose actual colors since they are more easily distinguishable.

Data Management >Meta-Network Transform

Aggregating Networks in ORA. Like UCINET and Pajek, ORA allows users to aggregate individual networks into a single-valued network. To do so, first highlight the meta-network you intend to aggregate in the Meta-Network Manager panel. Then, select the *Data Management> Meta-Network Transform* command, which calls up a dialog box similar to Figure 4.45. Next, select the "Create a new meta-network and then apply the transform" and "Combine into meta-network with one node class and one network" options. The former option is selected so that ORA does not overwrite the original network; the latter option is chosen in order to aggregate all of the networks in the meta-network into a single network. Finally, click on the "Transform" button, and a new meta-network will be generated that aggregates all of the networks contained in the meta-network you originally selected for aggregation.

4.9 Extracting and Simplifying Networks

Finally, let us consider a few techniques for simplifying networks and extracting subnetworks from the larger network. When working with large social networks, it is sometimes hard to make sense of the pattern of ties. In such situations, being able to shrink a network or extract a subset of a network can aid the analysis process. In particular, these techniques can be useful when you want to examine a subset of a network or to see whether there are patterns among the data that are not immediately observable when the network is looked at in its entirety.

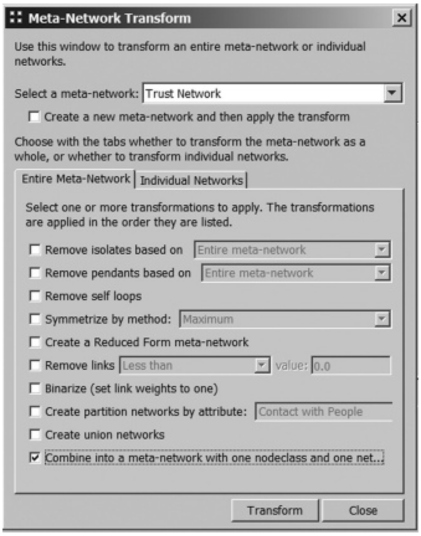


Figure 4.45. Transform Meta-Network Dialog Box (ORA)

Extracting Networks in UCINET

[UCINET] Data >Subgraphs from partitions

UCINET includes a tool for extracting subnetworks from larger networks: the *Data>Subgraphs from partitions* command. As its name suggests, you use a partition in order to identify the set or class of actors you want to extract from the network. This command brings up a dialog box similar to Figure 4.46. Note that we have to supply UCINET with both a network (`Operational Network (Aggregated).##h`) and a partition (`Attributes.##h`). In this particular case, we are extracting three subnetworks based on the current status attribute, which is located

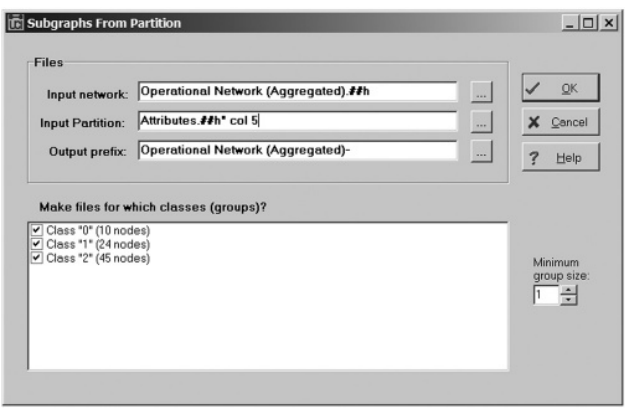


Figure 4.46. UCINET Subgraphs from Partition Dialog Box

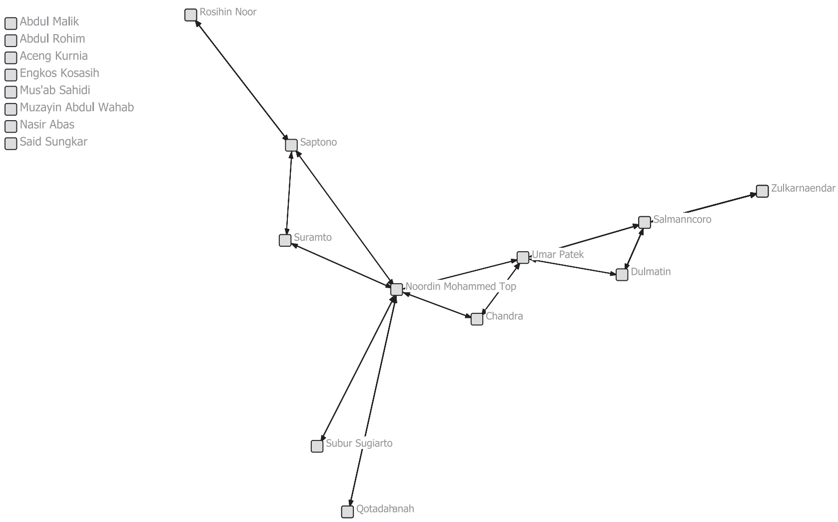


Figure 4.47. NetDraw Graph of Noordin Top's Alive and Free Operational Network

in the fifth column of the attribute partition (note that the fifth column is designated in the "Input Partition" box) and identifies which actors are dead (Class "0," 10 actors), which ones are alive and free (Class "1," 24 actors), and which ones are alive but in jail (Class "2," 45 actors; see Appendix 1). If we were only interested in obtaining the alive and free network, then we would have unchecked the "0" and "2" boxes. Clicking "OK" produces the three separate networks.

Figure 4.47 displays a NetDraw generated graph of Noordin's alive and free operational network. It indicates that Noordin sits at the center of this network and that several of its members have no ties with one another. Moreover, there are eight individuals who appear to be isolated from the network (located in the upper left of the graph), suggesting that they may no longer play a role in its operations. Interestingly, Nasir Abas, who was once the head of one of the local affiliates of Jemaah Islamiyah, the terrorist group from which Noordin's network emerged, now works with Indonesian authorities to get other terrorists to leave their violent pasts behind and rejoin Indonesian society (Mydans 2008). Unfortunately, in UCINET (and ORA) we cannot currently extract more than one type of actor into a single subnetwork (e.g., a network that includes everyone who was alive, regardless of whether they were free or not). We can do this in Pajek, however.

Simplifying (Collapsing) Networks in NetDraw

NetDraw does not include a tool for extracting subnetworks, but it does have one that allows users to collapse (i.e., shrink) networks based on

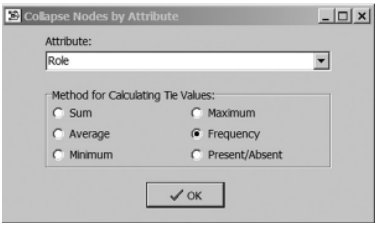


Figure 4.48. NetDraw Collapse Nodes by Attribute Dialog Box

a particular attribute (e.g., nationality, role, group membership). When collapsing a network, we shrink all of the actors that share a particular characteristic to a single new node. This type of analysis is something we might choose to do when we are interested in examining the pattern of ties between types of actors rather than the pattern of ties between the actors themselves. This is what some call a "global view" of the network (de Nooy et al. 2005:39–41), and it might illuminate patterns that were not detectable at the "street-level view," so to speak.

To collapse a network in NetDraw, first read the aggregated Noordin operational network file (`Operational Network (Aggregated.##h`) into NetDraw either with the *File>Open>Ucinet dataset>Network* command or by clicking on the open folder icon. Next, open the related attribute file (`Attributes.##h`) using the *File>Open>Ucinet dataset>Attribute data* command. If the attribute data do not match up with the network data (i.e., they have a different number of nodes), then you will receive a warning from NetDraw to that effect. If the data do match up correctly, you will receive no notice at all. If you want to examine the attribute data, you can access NetDraw's attribute editor (not shown) with its *Transform>Node attribute editor* command. Next, select the *Transform>Collapse Nodes by attribute* command, which brings up a dialog box similar to Figure 4.48. Here, we have selected "Role" as the characteristic/attribute on which to collapse the network. As you can see, NetDraw provides several options for calculating tie values between the collapsed nodes. In this case, we have selected "Frequency." Clicking "OK" completes the process and yields a network drawing similar to Figure 4.49.

File>Open >Ucinet dataset >Network

[NetDraw] File>Open >UCINET dataset >Network

Transform >Node attribute editor

Transform >Collapse Nodes by attribute

NetDraw (and Pajek and ORA) does not know the meaning of the various types of roles within the role attribute file, so it cannot assign meaningful labels to collapsed nodes. Instead, it places the letter "G" in front of all of the types of roles along with the number corresponding to that particular role. In Figure 4.49 the labels have been changed using NetDraw's *Properties>Nodes>Labels>Text* command. Under the label column at the bottom of the resulting dialog box (not shown), the various labels assigned to each role by NetDraw can be found. There, the nodes

Properties >Nodes>Labels >Text

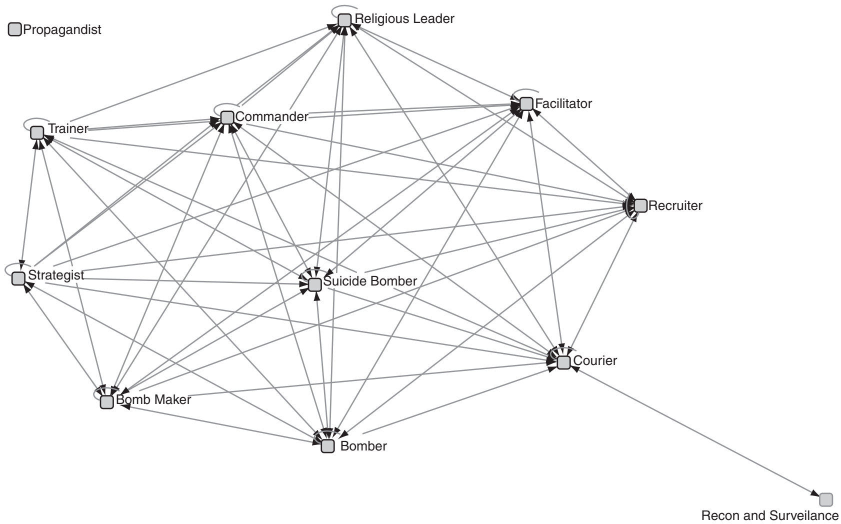


Figure 4.49. NetDraw Graph of Noordin's Operational Network Collapsed by Role

were relabeled using the codebook (see Appendix 1), which indicates which number is assigned to which role.

Extraction and Simplification in Pajek

Extracting Networks in Pajek. First, let's see how to extract a multiclass subnetwork in Pajek. Specifically, we will extract those actors within the operational network who are alive and either free or in jail. To do this we first have to load the Noordin network project file (`Noordin Network.paj`) using the *File>Pajek Project File>Read* command. Then, we need to ensure that the "Operational Network (Aggregated)" is showing in the Network drop-down menu and the "Current Status (ICG Report)" partition is showing in the Partition drop-down menu. Next, under the *Operations* menu, select the *Extract Network>Partition* command. A dialog box will appear that asks you which clusters (i.e., which classes) to select. Because we want those who are alive, choose clusters "1" and "2." When you click "OK," Pajek produces a new network and a new partition both of which contain sixty-nine actors.

[Pajek] [Main Screen] File>Pajek Project File>Read

Operations Extract Network >Partition

Now, let's create an education-level vector that matches the size of this new "alive" network. Make sure that the education-level vector is showing in the Vector drop-down menu. This vector contains the education level of the seventy-nine individuals included in the Noordin network. Next, make sure the current status partition is displayed in the Partition drop-down menu. Then, under the *Vector* menu, select

Vector >Extract Subvector

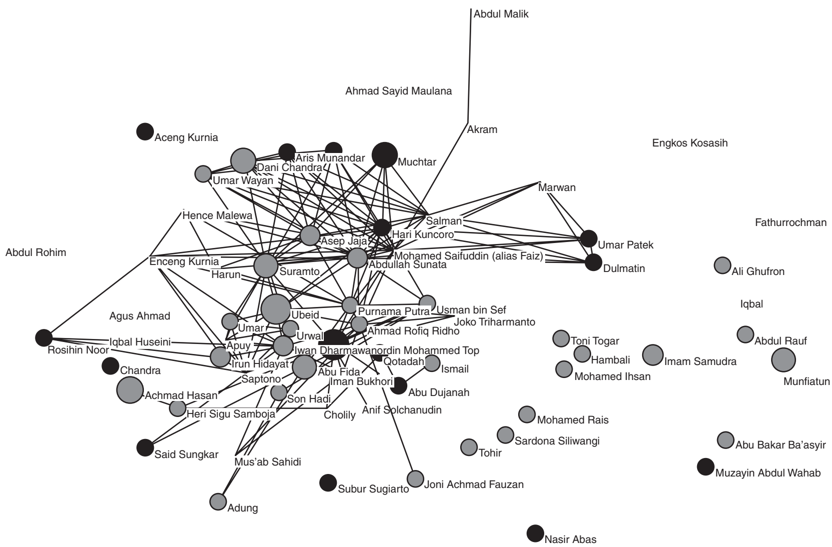


Figure 4.50. Pajek Drawing of Noordin Top's Alive Operational Network

the *Extract Subvector* command. A dialog box will appear, and like before, choose clusters "1" and "2" and click "OK." Pajek will create a new education vector with only sixty-nine actors. Finally, after making sure that the "Alive" Network, the new current status Partition, and the new Education Vector are all showing in their respective drop-down menus, select the *Draw-Partition-Vector* command under the *Draw* menu. Energize the drawing, and you should see a drawing similar to Figure 4.50 where the node colors indicate whether they are alive and free (black) or in alive and in jail (gray), and the node size indicates each individual's education level. Note that the nodes of some individuals are nonexistent. This is because they have an education level of "0." Note also that higher education level appears to be somewhat associated with the centrality of an individual in the network. In Chapter 7, we will see how to test to see if this is, in fact, true.

Draw>Draw-Partition-Vector

You can change Pajek's default colors with the *Options>Colors>Partition Colors* command. This calls up a dialog box (not shown) that contains forty squares that are colored and contain the partition's class numbers with which they are associated. For Figure 4.50, the "Default GreyScale 1" option has been chosen. If you click on the "Default Partition Colors" button, Pajek resets the original colors. If you want to change the color of a particular class, click on the square that contains the desired color and type in the number of the class with which you want to associate this color, and Pajek will swap the colors.

[Draw Screen] Options >Colors >Partition Colors

Simplifying Networks In Pajek. Shrinking (i.e., collapsing) networks in Pajek is straightforward. To illustrate how it is done, let's replicate what we did previously using NetDraw. That is, let's collapse Noordin's operational network by role. At Pajek's main screen, make sure that the "Operational Network (Aggregated)" is showing in the Network drop-down menu and the roles partition is showing in the Partition drop-down menu. Then, under the *Operations* menu, select the *Shrink Network>Partition* command. The first dialog box to appear asks you for the minimum number of connections between clusters. This refers to the minimum number of ties that must exist between pairs of shrunk nodes in order for there to be a tie between them. Generally, you will want to choose "1," which is the default. In the second dialog box, you can choose a class of actors not to be shrunk if that is what you want to do. In most cases, however, you will probably want to shrink all of the classes, so you need to choose a partition number that is not used in the partition you are using to shrink the network. In this instance, you will not want to choose "0" (Pajek's default) because "0" is a potential class in the current partition (see Appendix 1). Instead, choose "999." When you click "OK," Pajek shrinks all classes except the selected class and adds the shrunk network and a corresponding partition to the Network and Partition drop-down menus, respectively.

[Main Screen] Operations >Shrink Network >Partition

Like NetDraw, Pajek does not know the labels of the shrunken nodes, so it chooses the label of the first actor of a particular shrunken class that is shrunk and adds a "#" sign. If there is only one actor associated with a particular class (e.g., propagandist), it lists the actor's name without a "#" sign. In Pajek you can change the labels by manually editing the shrunk partition using the *File>Partition>Edit* command (of course, the shrunken partition has to be showing in the Partition drop-down menu before you select this command). One additional edit has been done to produce Figure 4.51. Because the "propagandist" and "recon and surveillance" roles each include only one actor, they appear as outliers when the network is visualized (see Figure 4.49). Thus, they were removed using the *Operations>Extract Network>Partition* command, where in the "Select Clusters" dialog box (not shown) the numbers "1–5,7–8, 10–12" were entered (the "recon and surveillance" cluster was "6," while the propagandist cluster was "9"). This graph appears to show that three roles are more central to Noordin's network (bomber, strategist, and trainer) than are the others (Noordin is classified as a strategist). This suggests that if Noordin were to be killed or captured,[11] then there is a strong probability that the network's future leadership would come from one of these more central roles.

File >Partition >Edit

Operations >Extract Network >Partition

[11] Noordin actually was killed in September 2009 by Indonesian authorities (International Crisis Group 2010).

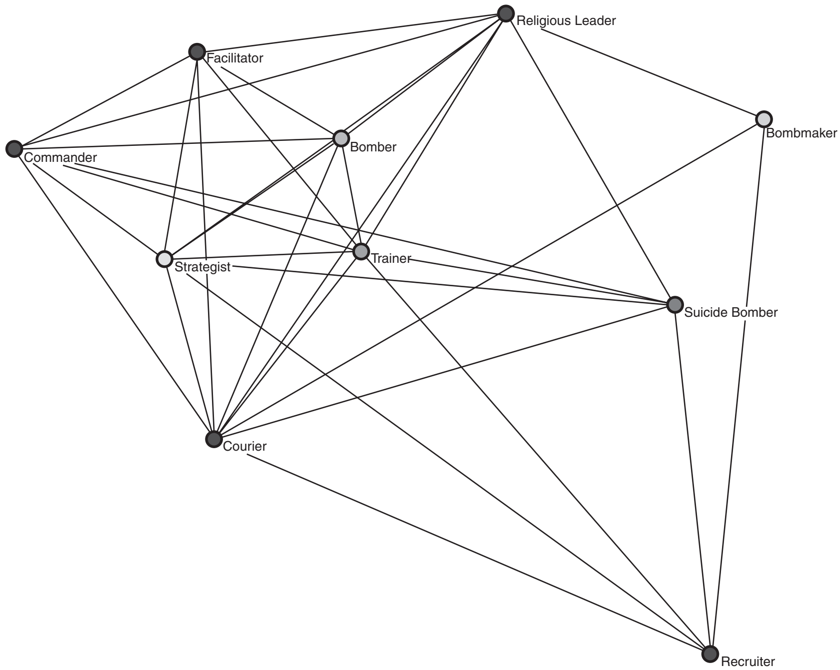


Figure 4.51. Pajek Drawing of Noordin's Operational Network Collapsed by Role

Extraction and Simplification in ORA

Extracting Networks in ORA. In ORA, open Noordin's trust network (`Trust Network.xml`) with its *File>Open Meta-Network* command. This meta-network contains the classmates, friendship, kinship, soulmates networks along with the associated attributes. To extract subnetworks in ORA, we use the *Attribute Partition Tool*, which is found under the *Data Management* menu. This calls up a dialog box similar to Figure 4.52. Here,we will extract only those members of the trust network who are incarcerated. To do this, we first have to select the trust network, then the current status attribute, and finally the attribute number (i.e., "2") that indicates whether someone is in jail or not. Clicking on the "Partition" button produces a new meta-network of those individuals who are alive but in jail (Figure 4.53). The isolates have been hidden using the *Actions>Isolates>Hide Isolate Nodes* command.

[ORA – Main Screen] File >Open Meta-Network

Data Management >Attribute Partition Tool

[Draw Screen] Actions >Isolates >Hide Isolate Nodes

Figure 4.53 indicates that some members of Noordin's incarcerated trust network are more central than are others. And we know from previous research that individuals on the periphery of a network are more likely to defect than are those at its center (Popielarz and McPherson 1995; Stark and Bainbridge 1980). This suggests that strategies that seek

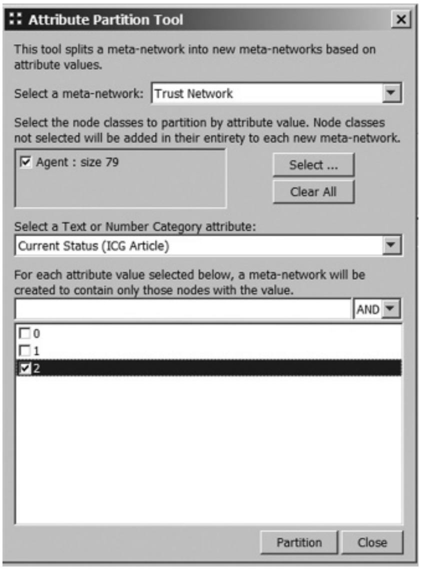


Figure 4.52. ORA Attribute Partition Tool

to rehabilitate captured terrorists (Mydans 2008) will probably be more effective if they focus their efforts on those on the periphery of the network than on those at the center.

Visualizations >View Networks >2D Visualization

Simplifying Networks in ORA. To shrink/collapse a network in ORA, highlight it in ORA's main screen Meta-Network Manager panel and then click the "Visualize" button or select the *Visualizations>View*

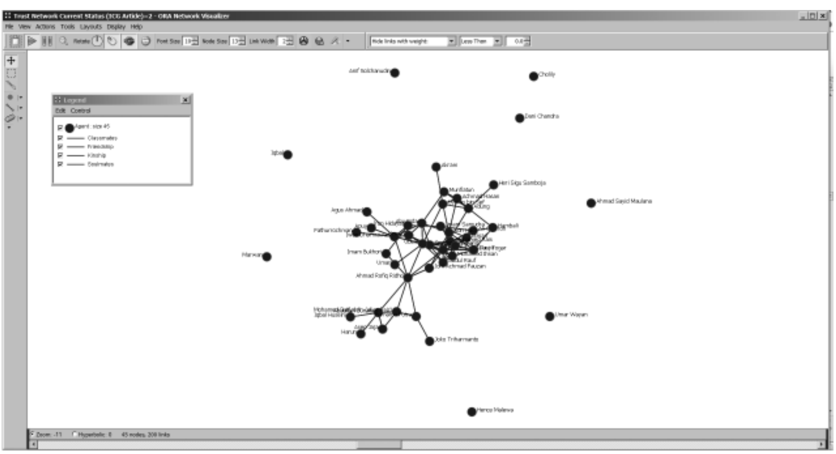


Figure 4.53. ORA Drawing of Noordin's Incarcerated Trust Network

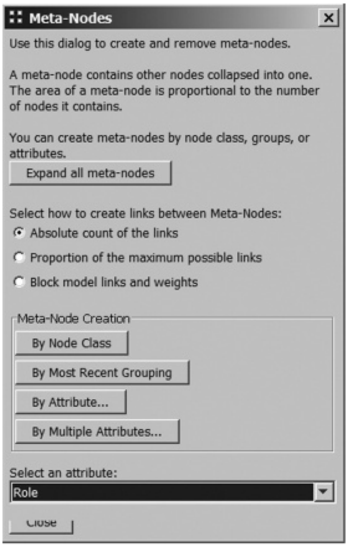


Figure 4.54. ORA's Meta-Node Manager

Networks>2D Visualization command. At the draw screen, choose the *Tools>Meta-Nodes* command, which calls up the Meta-Node dialog box (Figure 4.54). Select the "Absolute count of the links" options, and in the *Select an attribute* drop-down menu, indicate the attribute by which you want to collapse/shrink the network. Click on "Close" and ORA generates a collapsed network.

Tools>Meta-Nodes

Like Pajek and NetDraw, ORA does not know what to call the collapsed nodes, but it does allow us to rename the labels of the "meta-nodes," although it takes a few steps. First, we need to add the collapsed network to the meta-network using the draw screen's *File>Add Meta-Nodes to Current Meta-Network* command. This command creates a new node class (probably called "Meta Node 1: size 12") and two new networks in the Meta-Network Manager in the main screen. Select the new node class and click on the Editor tab located in the Network Information/Editor panel of the main screen. Two columns should be visible, one labeled "Node ID," the other labeled "Node Title." Under the latter, change the label names in the individual cells similar to the way we did it in NetDraw and Pajek. After doing this, select the "Meta Node 1 X Meta Node 1" network and click the "Visualize Only This Network" button, which will generate a new network map that should look similar to Figure 4.55. In the case of Noordin's trust network, strategists are the most central, which is not surprising considering that Noordin is a strategist. Nevertheless, other roles appear to be important in terms of Noordin's trust network, indicating that they may be crucial in terms of

File>Add MetaNodes to Current Meta-Network

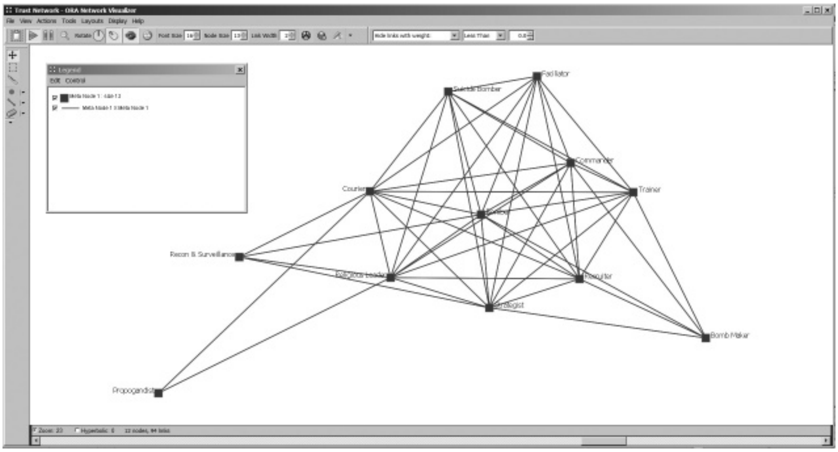


Figure 4.55. ORA's Meta-Node Network Map

the network's longevity (e.g., recruiting new members, preventing defections, etc.).

File>Save Meta-Network As...

Finally, it is worth noting that like Pajek, ORA lets you save all the files you have imported or created into a single "project file," which ORA calls a Meta-Network, with its *File>Save Meta-Network As*... command.

4.10 Summary and Conclusion

This chapter has focused on methods for collecting and recording social network data. It began with a brief discussion on how social network analysts define the boundaries of the networks they are examining. It then examined the difference between ego and complete networks, noting that social network analysts generally use complete networks when applying social network methods. Next, we considered the different types of data social network analysts use: one-mode, two-mode, and attribute data. We then discussed the various ways that social network analysts collect network data before looking at how social network data are recorded in matrix form. We then examined various methods for manipulating network data: deriving one-mode networks from two-mode networks; combining, aggregating, and parsing networks; and extracting and simplifying networks. It is now time to turn our attention to the various families of metrics that analysts use to examine social networks. We begin with methods for tapping into the various aspects of network topography.