

ORA 2006: User's Guide

ORA | Organizational Risk Analyzer

CASOS Technical Report¹

Kathleen M. Carley and Matt DeReno

August 2006
CMU-ISRI-06-113

Carnegie Mellon University
School of Computer Science
Institute for Software Research International (ISRI)
Center for Computational Analysis of Social and Organizational Systems (CASOS)

Abstract

ORA is a network analysis tool that detects risks or vulnerabilities of an organization's design structure. The design structure of an organization is the relationship among its personnel, knowledge, resources, and tasks entities. These entities and relationships are represented by the Meta-Matrix. Measures that take as input a Meta-Matrix are used to analyze the structural properties of an organization for potential risk. ORA contains over 50 measures which are categorized by which type of risk they detect. Measures are also organized by input requirements and by output. ORA generates formatted reports viewable on screen or in log files, and reads and writes networks in multiple data formats to be interoperable with existing network analysis packages. In addition, it has tools for graphically visualizing Meta-Matrix data and for optimizing a network's design structure. ORA uses a Java interface for ease of use, and a C++ computational backend. The current version ORA1.2 software is available on the CASOS website:

<http://www.casos.ece.cmu.edu/projects/ORA/index.html>.

¹ This work was supported by the ONR N00014-06-1-0104, the AFOSR for "Computational Modeling of Cultural Dimensions in Adversary Organization (MURI)", the ARL for Assessing C2 structures, the DOD, and the NSF IGERT 9972762 in CASOS. Additional support was provided by CASOS and ISRI at Carnegie Mellon University. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the National Science Foundation, the Department of Defense, the Office of Naval Research, the Army Research Labs, the Air Force Office of Sponsored Research or the U.S. government.

Keywords: dynamic network analysis, measures, meta-matrix, organization risk

ORA User's Guide Table of Contents

ORA Organizational Risk Analyzer.....	1
Getting Started	8
What Is ORA?.....	9
An Overview.....	9
The ORA Visualizer	9
Reports.....	9
Charts.....	9
System Requirements.....	10
Basic Terms	11
Social Network Analysis (SNA).....	12
Where to find out more on SNA	163
Dynamic Network Analysis.....	13
Where to learn to more	13
ORA and DNA.....	14
Where To Begin.....	15
ORA's Main Interface	16
ORA Reports.....	17
Reports Explained.....	19
Risk Report.....	19
Intelligence Report	19
Management Report	19
Context Report	19
Subgroup Report	19
Sphere of Influence Report	19
Optimization Report.....	19
ORA Charts.....	20
Chart Types Explained.....	23
ORA Bar Chart.....	23
Scatter Plot	24
Histogram	25
Heat Map.....	26

The Network Optimizer	27
Optimization Method	29
Default and Advanced Experiment Type.....	31
Monte Carlo Network Optimization	33
Simulated Annealing Network Optimization.....	36
Over-Time Viewer	39
The ORA Visualizer	40
Visualizer Tools	41
Drill Down Wizard	43
Drill-Down Wizard Explained.....	46
Drill Down Wizard Example	48
Creating A MetaNode	55
The Path Finder.....	58
Path Finder Example.....	60
Sphere of Influence	64
Sphere of Influence Window.....	66
Sphere of Influence Example.....	67
Node Status	73
Node Status Example.....	74
Actions	76
Compute FOG Groups	77
Compute FOG Group Example	79
ORA Measures.....	83
The Bonacich Power Centrality	86
Centrality, Eigenvector	86
Centrality, Total Degree.....	86
Access Index, both Knowledge and Resource Based.....	87
Access Index, both Knowledge and Resource.....	87
Boundary Spanner.....	87
Centrality, Betweenness.....	89
Centrality, Bonacich Power	89
Centrality, Closeness.....	90
Centrality, Eigenvector	90
Centrality, In Degree.....	90
Centrality, Information	91
Centrality, Inverse Closeness.....	91
Centrality, Out Degree	91
Centrality, Total Degree.....	92
Clique Count	92
Clustering Coefficient, Watts-Strogatz.....	92
Cognitive Demand	94
Cognitive Distinctiveness	94
Relative Cognitive Distinctiveness	94
Cognitive Expertise.....	96
Relative Cognitive Expertise	96
Cognitive Resemblance	96

Cognitive Similarity.....	97
Relative Cognitive Similarity	97
Communication.....	97
Component Count, Strong	98
Component Count, Weak.....	98
Component Members, Weak	98
Congruence, Agent Knowledge Needs.....	99
Congruence, Agent Resource Needs.....	99
Congruence, Agent Knowledge Waste	99
Congruence, Agent Resource Needs.....	100
Congruence, Communication	100
Congruence, Organization Agent Knowledge Needs	100
Congruence, Organization Agent Resource Needs.....	101
Congruence, Organization Agent Knowledge Waste	101
Congruence, Organization Agent Resource Waste.....	101
Congruence, Organization Task Knowledge Needs	102
Congruence, Organization Task Resource Needs.....	102
Congruence, Organization Task Knowledge Waste	102
Congruence, Organization Task Resource Waste.....	103
Congruence, Strict Knowledge	103
Congruence, Strict Resource.....	103
Congruence, Task Knowledge Needs	104
Congruence, Task Resource Needs.....	104
Congruence, Task Knowledge Waste	104
Congruence, Task Resource Waste.....	105
Connectedness, Krackhardt.....	105
Constraint, Burt.....	105
Density	106
Diameter.....	106
Distance Weighted Reach.....	106
Diversity, Knowledge	107
Diversity, Resource.....	107
Edge Count, Lateral	107
Edge Count, Lateral	107
Edge Count, Reciprocal	109
Edge Count, Sequential.....	109
Edge Count, Skip	109
Effective Network Size.....	110
Efficiency, Global	110
Efficiency, Krackhardt.....	110
Efficiency, Local.....	111
Exclusivity, Knowledge.....	111
Exclusivity, Resource	111
Exclusivity, Task.....	112
Fragmentation	112
Hierarchy, Krackhardt.....	112

Interdependence	113
Interlockers	113
Radials.....	113
Load, Knowledge.....	114
Load, Resource	114
Negotiation, Knowledge	114
Negotiation, Resource.....	115
Network Centralization, Betweenness	115
Network Centralization, Closeness.....	115
Network Centralization, Column Degree	116
Network Centralization, In Degree	116
Network Centralization, Out Degree	116
Network Centralization, Row Degree.....	117
Network Centralization, Total Degree.....	117
Network Levels.....	117
Node Levels	118
Omega, Knowledge.....	118
Omega, Resource	118
Performance as Accuracy	119
Personnel Cost	119
Potential Workload, Knowledge.....	119
Potential Workload, Resource	120
Redundancy, Access	120
Redundancy, Assignment	120
Redundancy, Column.....	121
Redundancy, Knowledge	121
Redundancy, Resource.....	121
Redundancy, Row	122
Shared Situation Awareness	122
Simmelian Ties	122
Span of Control.....	123
Speed, Average	123
Speed, Minimum.....	123
Task Completion, Knowledge Based.....	124
Task Completion, Resource.....	124
Task Completion, Overall.....	124
Task Completion, Overall.....	125
Transitivity.....	125
Triad Count	125
Under Supply, Knowledge.....	126
Under Supply, Resource	126
Upper Boundedness, Krackhardt	126
Tasks	127
Simplifying A Complex Visual Network	128
Working Inside The Visualizer.....	129
The Tool Bar Explained.....	130

Open Folder.....	130
Copy Paste.....	130
Play / Pause	130
Magnifying / Maximizing	131
Rotating The Visualization.....	131
Eliminating Labels	132
Removing Isolates.....	134
Removing Pendants	136
Creating A MetaNode	138
Zooming.....	141
Hyperbolic View	143
Rotating A Visualization.....	145
Adding And Removing Nodes.....	146
Adding and Removing Nodes In The Editor	147
Adding and Removing Nodes In The Visualizer	148
Adding and Removing Nodes in the Key Set	150
From the ORA Tool Bar > Window > Key Set Selector	150
Creating a MetaMatrix from an Excel Spreadsheet	153
Running An Over-Time Analysis	159
References.....	162
Additional Resources for CASOS tools and this tool chain:	163

Getting Started

Welcome to ORA's Help File system. The ORA help set is organized into the following top-level folders:

- Getting Started
- The Main Interface
- The Visualizer
- ORA Measures
- Tasks

Each top level folder contains increasingly detailed content relating to that topic.

What Is ORA?

An Overview

The Organizational Risk Analyzer (ORA) is a statistical analysis package for analyzing complex systems as dynamic social networks. Many complex systems such as organizations, intra-state alliances, food webs, etc. can be represented as an ecology of interlinked networks. Within ORA any complex system is represented as a MetaMatrix.

What follows is a general description of ORA's primary capabilities. Specific instruction is provided under the correlating folders in this help system.

The ORA Visualizer

The ORA Visualizer renders conceptual images of social networks. Entities such as Agents, Task, Knowledge and Organizations are represented as *nodes*. Nodes which share the same attributes can be further grouped together creating metanodes. Links, also called ties or edges, connect nodes that share a direct relationship. Such relationships are derived from the *MetaMatrix*, ORA's single unit of data input, and are referred to as graphs. The ORA visualizer is interactive; You can zoom, rotate, isolate, add and remove nodesets, and much more. See Basic Terms for additional definitions relating to Social Networks.

Reports

ORA can run many reports: Risk, Intelligence, and Sphere of Influence to name a few. Multiple organizations can be compared against each other, network structure can be optimized, subgroups within a network can be identified, and scenarios involving the removal of agents, links, or nodes can be examined. Reporting capabilities are constantly being refined and updated. See Advanced Usages for detailed instruction and an explanation of each of ORA's current reports.

Charts

Four chart types are available: Bar Chart, Scatter Plot, Histogram, and Heat Map. Each one in turn presents a different statistical profile of a selected nodeset. Examples of these reports and how to access them can be found under Basic Usages. See Advanced Usages for detailed description of the measures and algorithms that drive ORA Charts.

System Requirements

ORA performs best on machines that meet or exceed the following specifications:

Windows XP, 256 MB RAM, Pentium 4 processor (or equivalent) running at 1.0 GHz.

When working with extremely large data sets, increasing processing speed and RAM is highly recommended.

Basic Terms

Entity Class – The type of items we care about (e.g., actors).

Entities – General things within an entity class (e.g. a set of actors such as employees).

Node – A specific entity (e.g., Joe, Martha, Bob; or, airplanes, buses, bicycles).

Dyad – Two nodes and the connection between them.

Dyadic Analysis – Statistical analysis where the data is in the form of ordered pairs or dyads. The dyads in such in such an analysis may or may not for a network.

Relation – The way in which entities in one class relate to entities in another class.

Link – A specific relation among two nodes (also referred to as a connection, edge or tie).

Network – Set of links among nodes. Nodes may be drawn from one or more entity classes and links may be of one or more relation classes.

MetaMatrix – A statistical graph of correlating factors of personnel, knowledge, resources and tasks. These measures are based on work in social networks, operations research, organization theory, knowledge management, and task management.

Multiplex – Network where the links are from two or more relation classes.

Multimode Network – Where the nodes are in two or more entity classes.

Node Set – A collection of nodes that group together for some reason.

Classic SNA density – The number of edges divided by the number of possible edges not including self-reference. For a square matrix, this algorithm first converts the diagonal to 0, thereby ignoring self-reference (a node connecting to itself) and then calculates the density. When there are N nodes, the denominator is $(N*(N-1))$. To consider the self-referential information use general density.

General density – The number of edges divided by the number of possible edges including self-reference. For a square matrix, this algorithm includes self-reference (a node connecting to itself) when it calculates the density. When there are N nodes, the denominator is $(N*N)$. To ignore self-referential information use classic SNA density.

Social Network Analysis (SNA)

Social Network Analysis is a scientific area focused on the study of relations, often defined as social networks. In its basic form, a social network is a network where the nodes are people and the relations (also called links or ties) are a form of connection such as friendship. Social Network Analysis takes graph theoretic ideas and applies them to the social world. The term "social network" was first coined in 1954 by J. A. Barnes (see: *Class and Committees in a Norwegian Island Parish*). Social network analysis is also called network analysis, structural analysis, and the study of human relations. SNA is often referred to as the science of "connecting the dots."

Today, the term Social Network Analysis (or SNA) is used to refer to the analysis of any network such that all the nodes are of one type (e.g., all people, or all roles, or all organizations), or at most two types (e.g., people and the groups they belong to). The metrics and tools in this area, since they are based on the mathematics of graph theory, are applicable regardless of the type of nodes in the network or the reason for the connections.

For most researchers, the nodes are actors. As such, a network can be a cell of terrorists, employees of global company or simply a group of friends. However, nodes are not limited to actors. A series of computers that interact with each other or a group of interconnected libraries can comprise a network also.

Dynamic Network Analysis

Dynamic Network Analysis (DNA) is an emergent scientific field that brings together traditional social network analysis (SNA), link analysis (LA) and multi-agent systems (MAS). There are two aspects of this field. The first is the statistical analysis of DNA data. The second is the utilization of simulation to address issues of network dynamics. DNA networks vary from traditional social networks in that are larger dynamic multi-mode, multi-plex networks, and may contain varying levels of uncertainty.

DNA statistical tools are generally optimized for large-scale networks and admit the analysis of multiple networks simultaneously in which, there are multiple types of nodes (multi-node) and multiple types of links (multi-plex). In contrast, SNA statistical tools focus on single or at most two mode data and facilitate the analysis of only one type of link at a time.

DNA statistical tools tend to provide more measures to the user, because they have measures that use data drawn from multiple networks simultaneously. From a computer simulation perspective, nodes in DNA are like atoms in quantum theory, nodes can be, though need not be, treated as probabilistic. Whereas nodes in a traditional SNA model are static, nodes in a DNA model have the ability to learn. Properties change over time; nodes can adapt: A company's employees can learn new skills and increase their value to the network; Or, kill one terrorist and three more are forced to improvise. Change propagates from one node to the next and so on. DNA adds the critical element of a network's evolution and considers the circumstances under which change is likely to occur.

Where to learn to more

Kathleen M. Carley, 2003, "Dynamic Network Analysis" in *Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*, Ronald Breiger, Kathleen Carley, and Philippa Pattison, (Eds.) Committee on Human Factors, National Research Council, National Research Council. Pp. 133-145, Washington, DC.

Kathleen M. Carley, 2002, "Smart Agents and Organizations of the Future" *The Handbook of New Media*. Edited by Leah Lievrouw and Sonia Livingstone, Ch. 12, pp. 206-220, Thousand Oaks, CA, Sage.

Kathleen M. Carley, Jana Diesner, Jeffrey Reminga, Maksim Tsvetovat, 2005-forthcoming, *Toward an Interoperable Dynamic Network Analysis Toolkit*, DSS Special Issue on Cyber infrastructure for Homeland Security: Advances in Information Sharing, Data Mining, and Collaboration Systems.

ORA and DNA

In general, you may want to use ORA in conjunction with other computational tools to advance DNA theory. The CMU CASOS tools that work with ORA to form tool chains are AutoMap (extracts networks from texts) and various DNA simulators including both Construct and DyNet. These tools have been used in a number of real world applications:

- Designing adaptive teams for Command and Control Networks
- Estimating the impact of organizational downsizing
- Estimating the effectiveness of new structures
- Evaluating risk in organizational designs
- Examine impact of IT effectiveness
- Impact analysis of actions in asymmetric warfare simulation
- Impact analysis of weaponized biological attacks on cities

ORA is interoperable with a number of other SNA and link-analysis tools: *UCINET*, *KeyPlayer*, and *Analyst Notebook*. Additional information is listed under data import and export.

Where To Begin

To begin, you must load a Meta-Matrix into ORA. Your Meta-Matrix file can be of the following file formats: DyNetML (a specially defined markup language for representing dynamic networks), CSV, GraphML, and the text (DL) and binary data formats of UCINET.

Other methods to load a Metamatrix include building your own graph from an excel spreadsheet and cutting and pasting the information directly into ORA. For now, we will assume you have a MetaMatrix in one of the commonly defined formats above. There are two ways to load a MetaMatrix:

1) Open the folder icon in the tool bar > Select the directory where the Metamatrix data set is saved; or,

2) From the drop down menu: File > Open Metamatrix > Select the Metamatrix from the appropriate directory.

ORA can work with multiple MetaMatrices. Whether you load a MetaMatrix for the first time or if you had been working with a MetaMatrix from a previous session, a pop-up window displays the following button options:



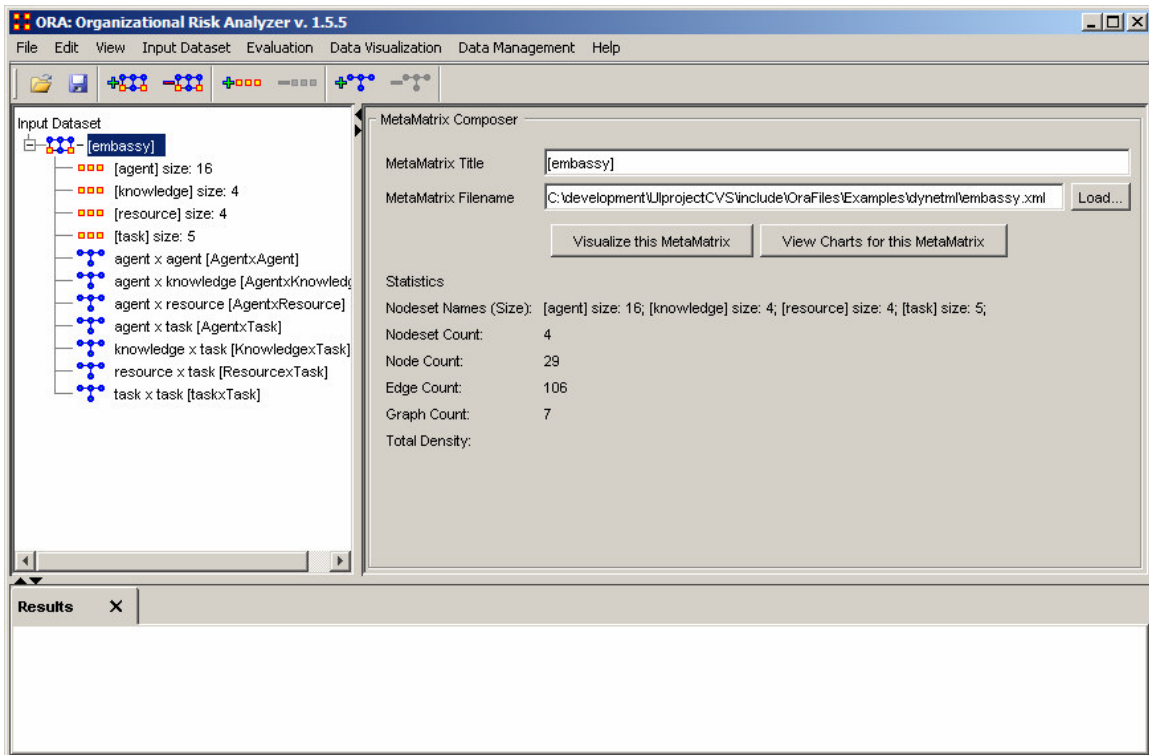
Choose Replace Selected Org to remove the previous Metamatrix and replace it with the current one; or, choose Append As Additional Org to add it to any MetaMatrices already loaded in the ORA session; Choose Replace all Orgs to remove any Metamatrices loaded into ORA and start over with the current Metamatrix only.

ORA's Main Interface

The ORA interface is organized into three resizable window panes. On the left corner appears a tree directory of current MetaMatrices loaded into ORA and associated subdirectories. On the right side, which typically loads much larger by default, appears the *MetaMatrix Composer*. This window pane allows quick access to the *ORA Visualizer* and Chart features. By clicking *Visualize this Metamatrix*, the currently loaded MetaMatrix will render in the ORA Visualizer.

Tip! Visualizing a MetaMatrix is a great way to become familiar with how the Visualizer interacts with the main interface.

Selecting the View Charts button loads the Charts Results window where the various chart types will be tabbed. Note the considerable amount of empty space in the ORA interface's bottom pane. This area will be used for report data generation. For now, it can be resized to one's preferences.



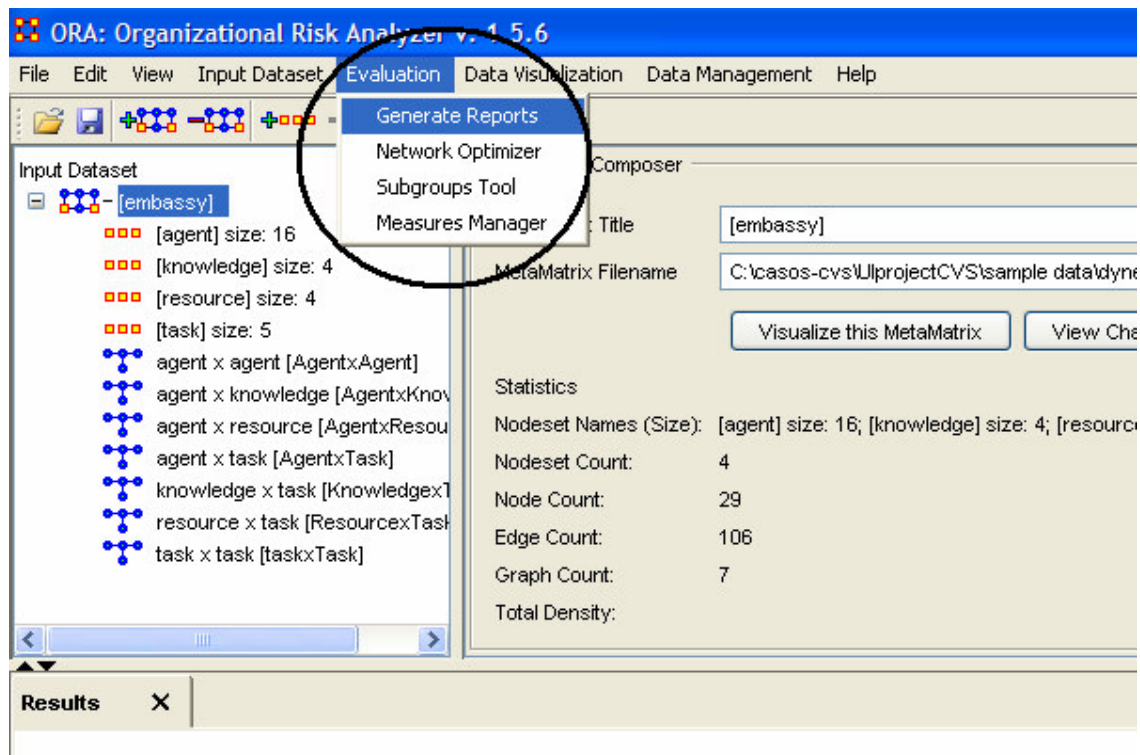
ORA Reports

Reports are one of ORA's core functions, which also include the Visualizer and Chart tools. The black circle on the screen shot below highlights how to access ORA Reports from the main interface.

From the drop down menu: Evaluation > Generate Reports

ORA Reports provide a computational tool to analyze data that make up a network. Reports give you the numbers behind the network. ORA reports are driven by a variety of key measures proven useful to researchers in the analysis of networks. These measures, and specific types of reports, will be covered in greater detail under Advanced Usages. For now, the goal is to be able to load a MetaMatrix, select a report to generate, and view the results. Scroll down for instruction and a series of screen shots illustrating how to run an ORA Report.

[General description of ORA's basic reports](#)



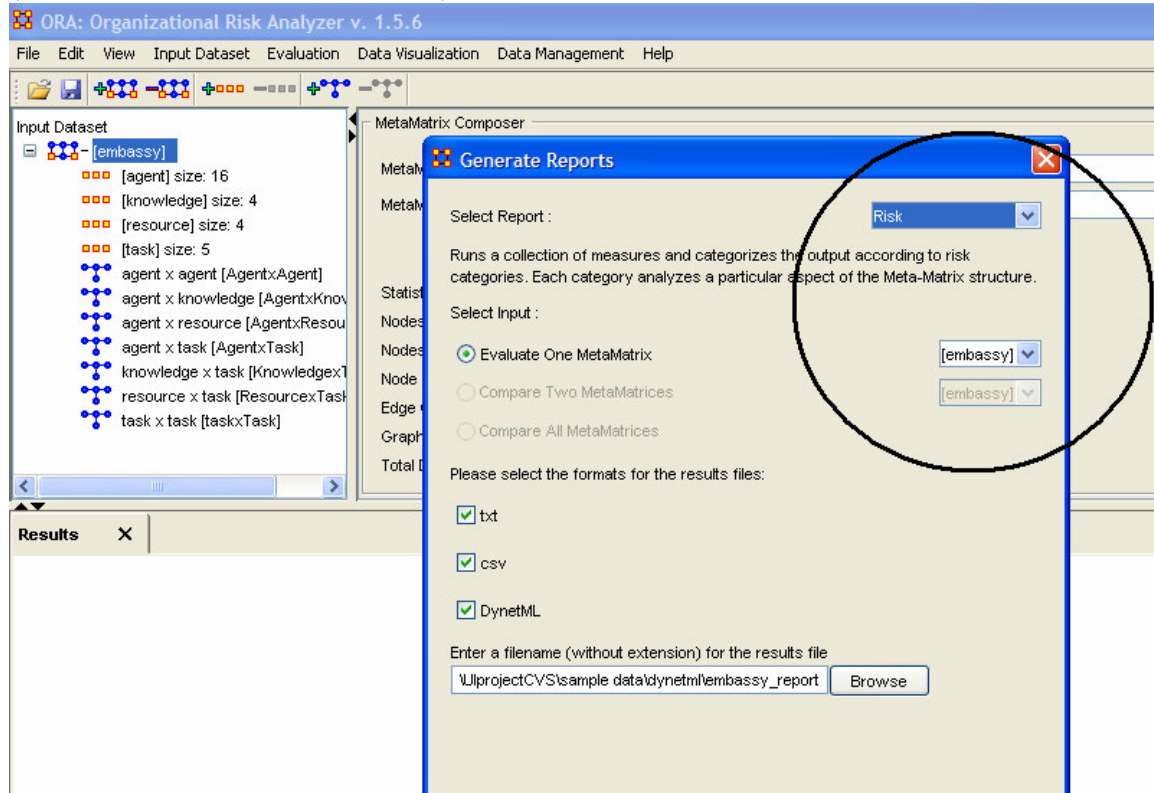
After selecting reports, the following pop up window should appear (see screen shot below). The black circle highlights a drop down menu area. Click the drop down tab to see the entire list of available reports.

From the drop down menu: Evaluation > Generate Reports > Select report type

This drop down tab lists available reports. As of ORA Version 1.5.8, there are seven reports available. More may be added in future versions. In this example, please note the

gray'd out options under Select Input. These options allude to one of the advanced features of ORA Reports, which is the ability to compare and analyze multiple networks simultaneously. This will be covered in greater detail under Advanced Usages.

(Note: More screen shots below...)



In this example, we will run the Risk report. To do so, click Finish. The screen shot below, reflects the results of running the Risk report on the MetaMatrix labeled Embassy.

From the drop down menu: Evaluation > Generate Reports > Select report type (Risk) > Finish = "Analysis Complete"

Note that the bottom window pane is now populated with data. This data is the result of successfully running the Risk report. Additionally, you will see a series of tabs at the top of this window pane. These tabs reflect additional reports and can be accessed by simply clicking on them. The details of these reports will be covered under Advanced Usages. At this point, you should be able to load a MetaMatrix and run a report based on it.

Reports Explained

Risk Report

Evaluates the overall system using measures of risk or vulnerability in seven different areas.

Intelligence Report

Identifies key actors individuals and groups – who by virtue of their position in the network are critical to its operation.

Management Report

Identifies over- and under-performing individuals and assesses the state of the network as a functioning organization.

Context Report

Compares measured values against various stylized forms of networks in an effort to characterize the networks topology.

Subgroup Report

Identifies the subgroups present in the network using various grouping algorithms.

Sphere of Influence Report

For each individual, identifies the set of actors, groups, knowledge, resources, etc. that influence and are influenced by that actor.

Optimization Report

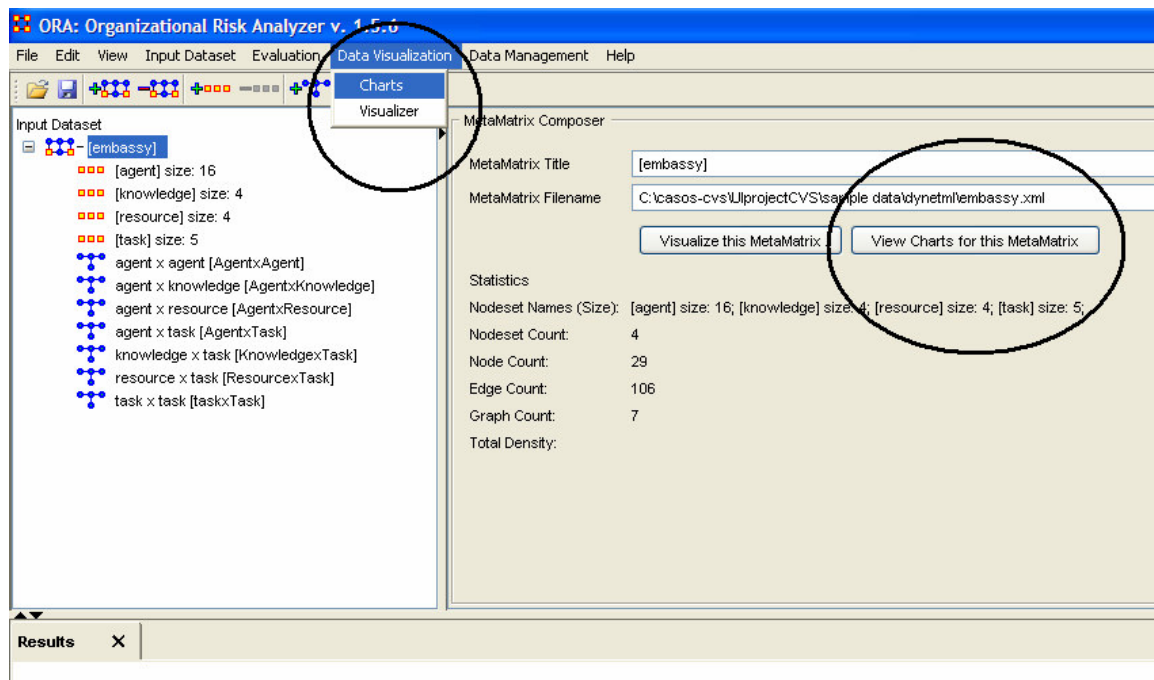
Enables the analyst to locate the optimal form of the target organization and/or assess how far the current structure is from the optimum.

ORA Charts

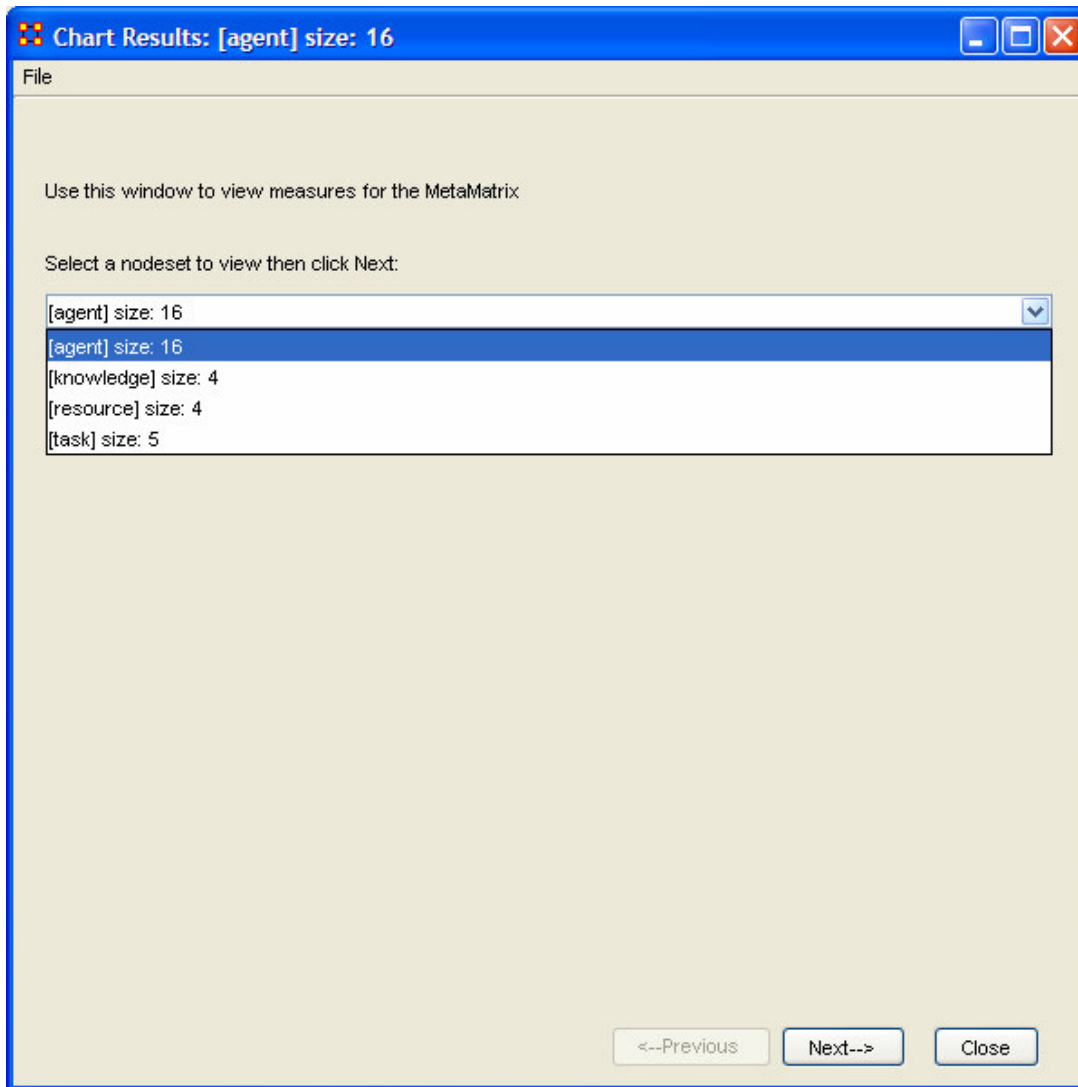
Four charts are available: The Bar chart, Scatter Plot, Histogram, and Heat Map. ORA Charts can be accessed through the Main Interface by clicking the View Charts For This MetaMatrix button in the MetaMatrix Composer window pane or from the Main Interface drop down menu.

From the drop down menu: Data Visualization > Charts

In the screen shot below, black ellipses highlight both methods to access ORA Charts. Scroll down for more instruction and screen shots on how to create a chart in ORA.



Using either method, ORA produces the following pop-up window.



Select the Nodeset you are interested in Charting and click the Next button. For this example, we will simply select the currently loaded nodeset Agents. ORA produces the following Bar Chart, displayed in the screen shot below.

By now, you should be able to access ORA Charts by loading a MetaMatrix by either using the Charts button in the MetaMatrix Composer window pain or from the drop down menu of the main interface. See Chart Types Explained for more detailed information about the four chart types.

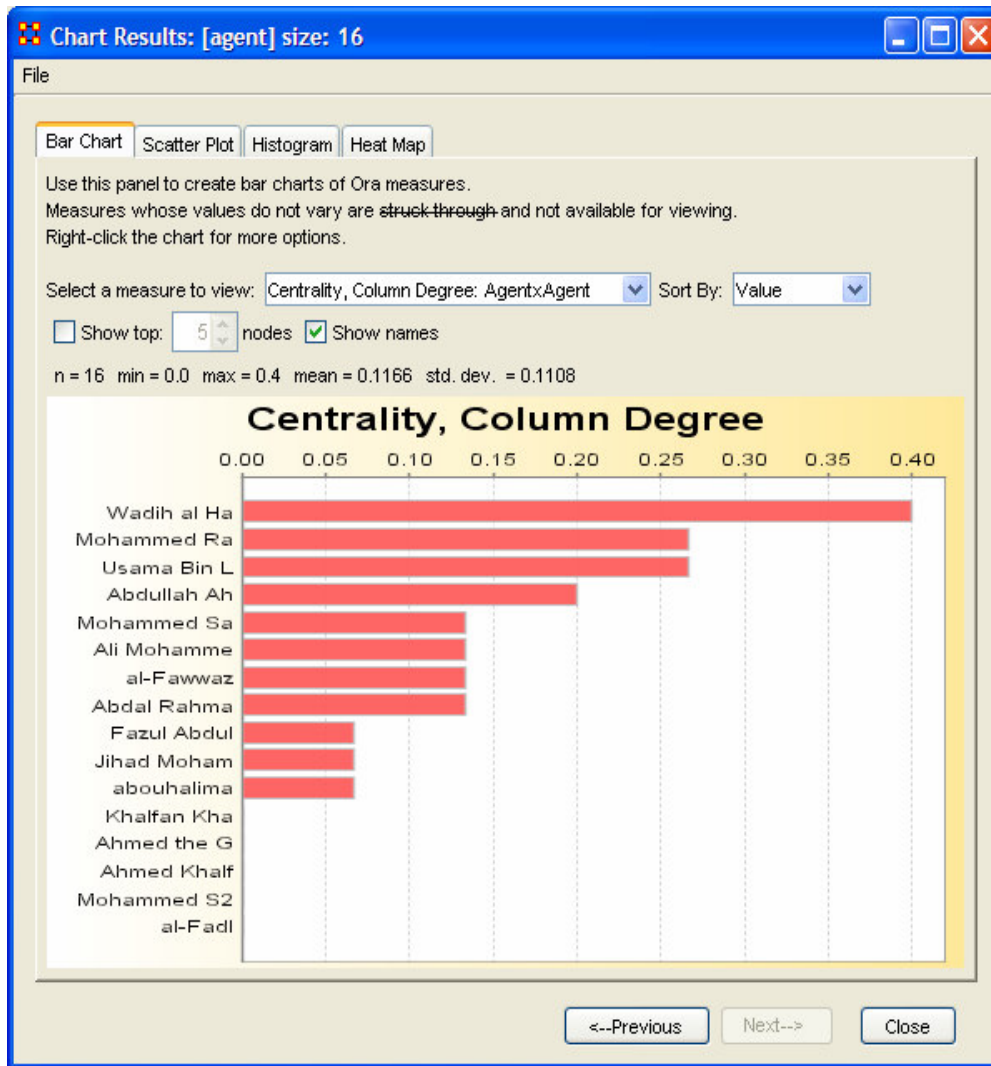
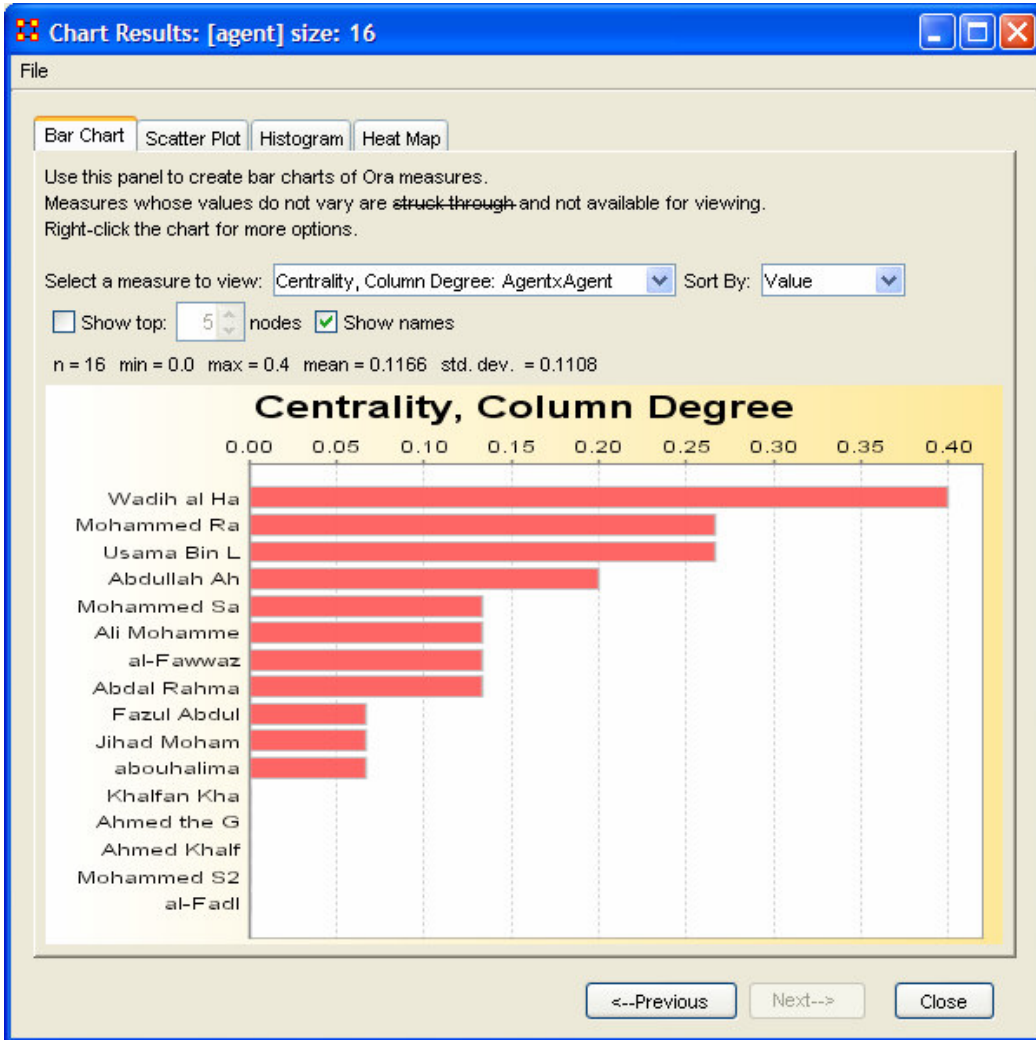


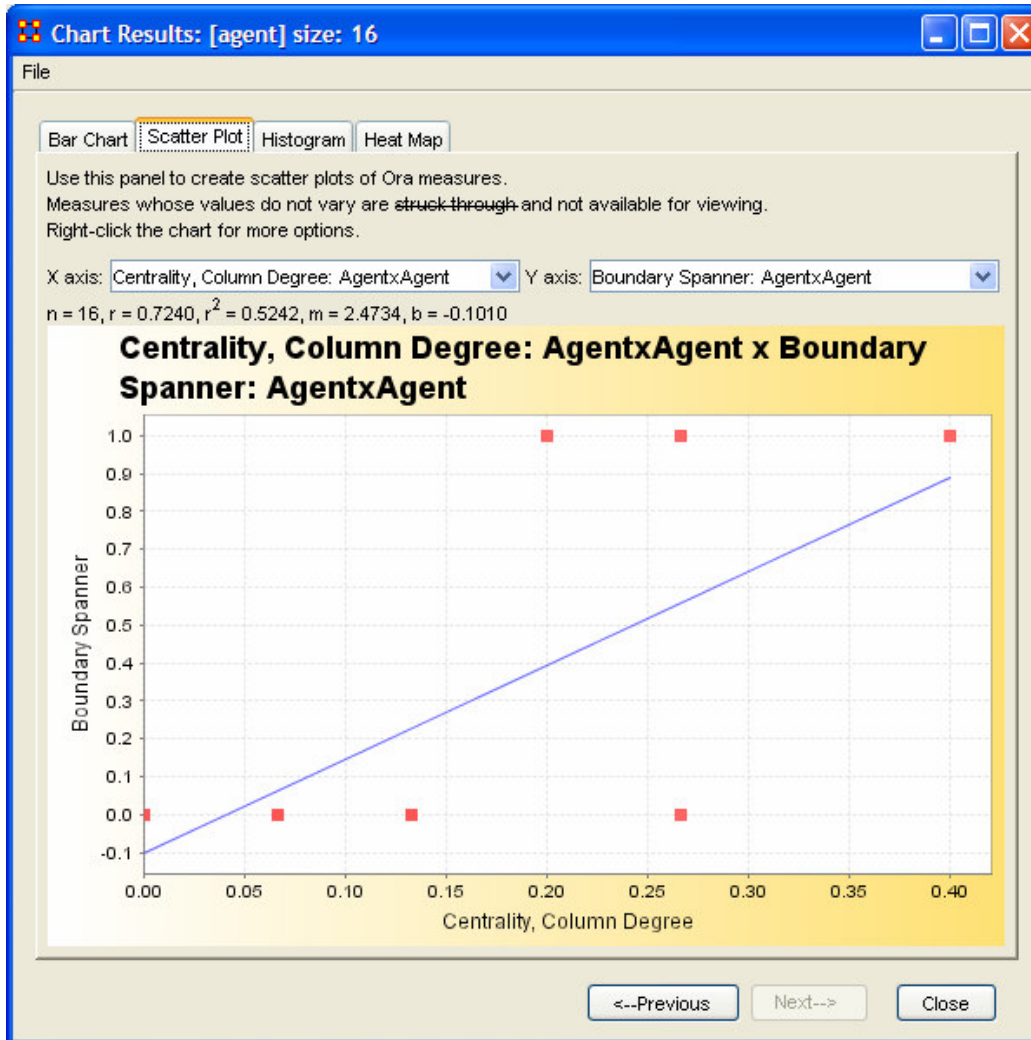
Chart Types Explained

Below are examples of each type of ORA charts: *Bar Chart*, *Scatter Plot*, *Histogram*, and *Heat Map*.

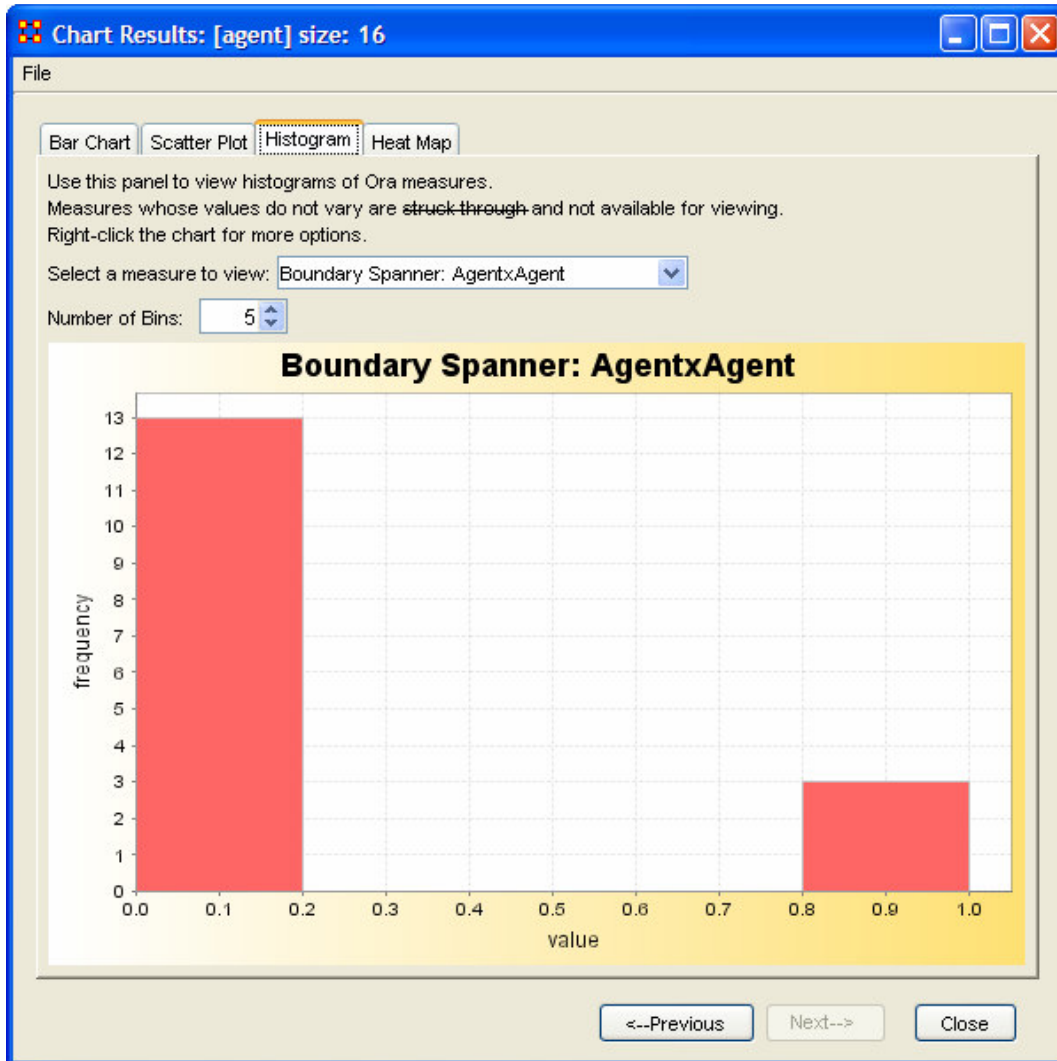
ORA Bar Chart



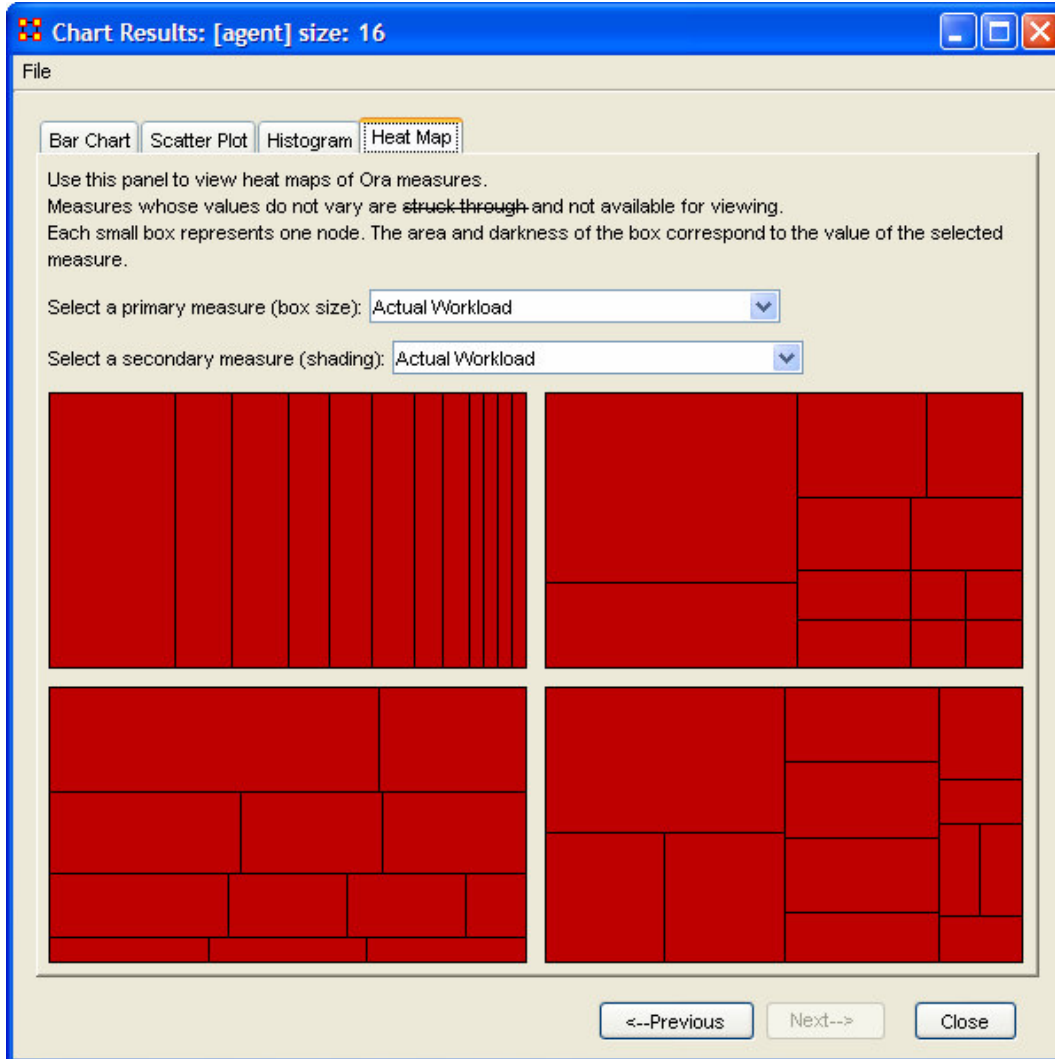
Scatter Plot



Histogram



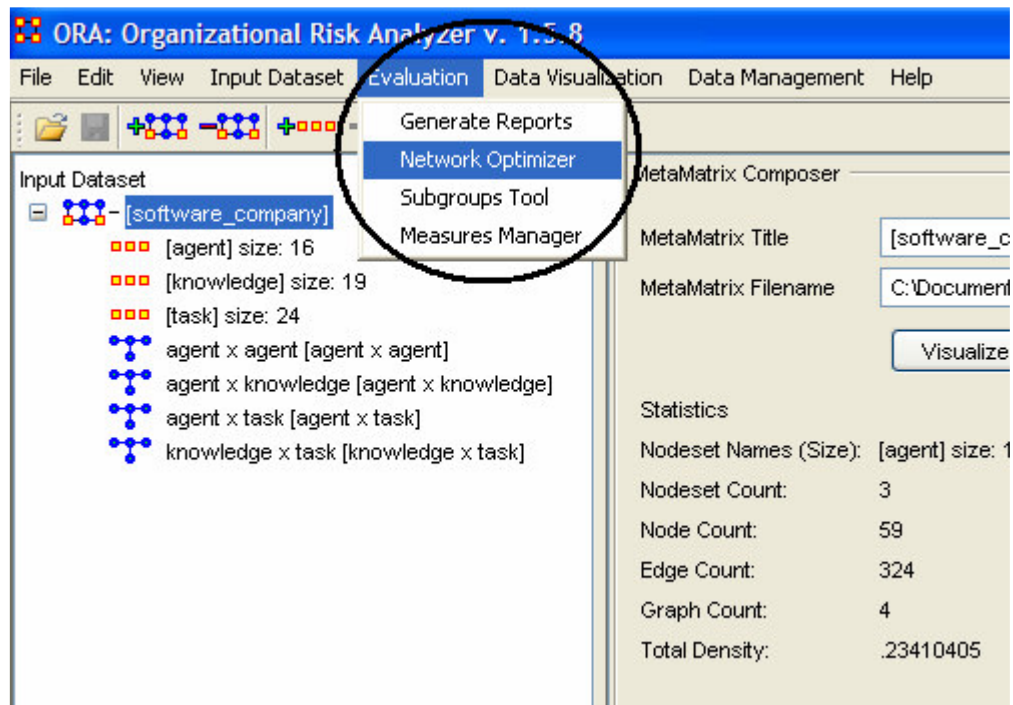
Heat Map



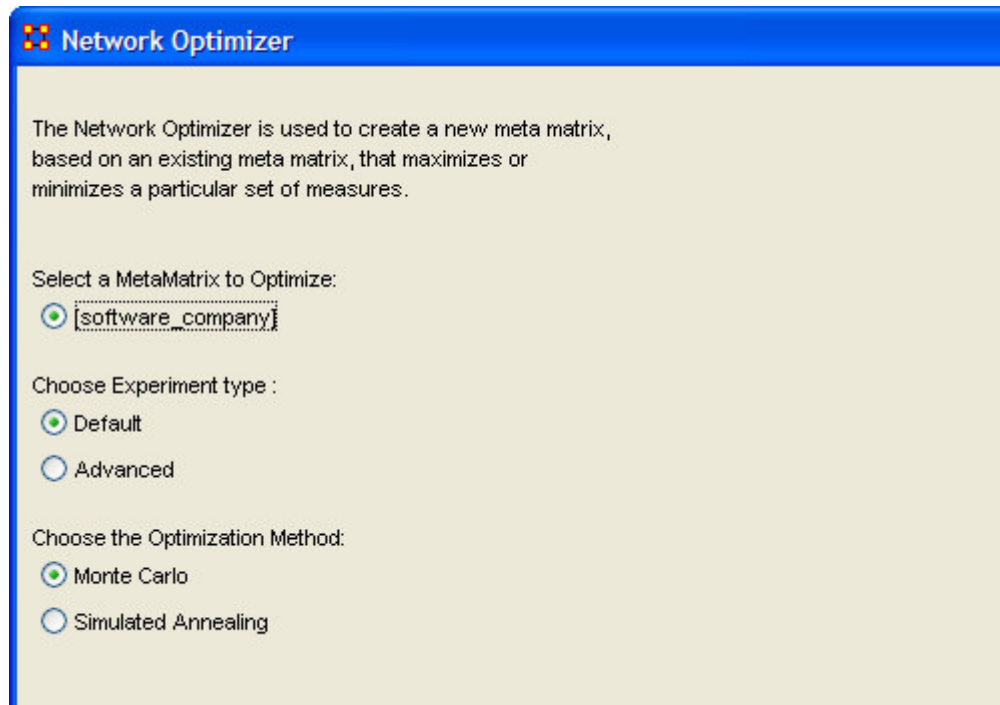
The Network Optimizer

The Network Optimizer is a tool to maximize one or more measures within your organization. You can use this tool to change network variables and analyze your organization under various scenarios. First you must have a MetaMatrix loaded.

From the drop down menu: Evaluation > Network Optimizer



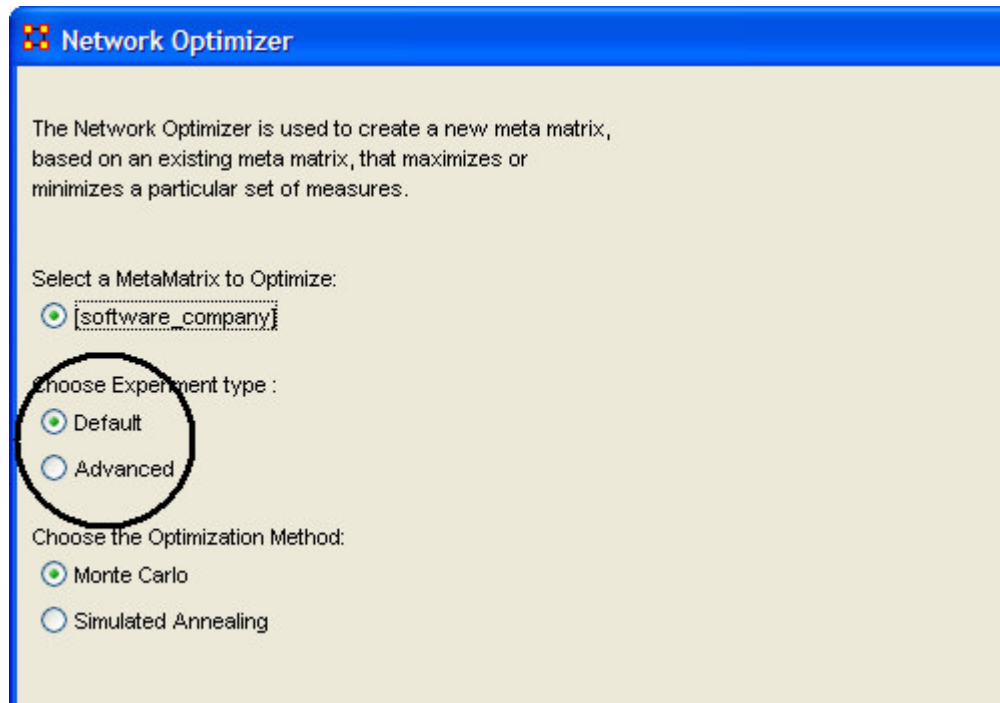
Selecting the Network Optimizer will bring up the Network Optimizer wizard interface. Below is a screen shot of the Network Optimizer Interface.



Select the organization you wish to optimize. You can choose from any organization you have loaded into the MetaMatrix Manager. In this example, we will use the MetaMatrix "software_company.xml."

Experiment Type

"Advanced" allows you to select various options. "Default" makes the selection for you. The black ellipse below shows where to make this selection.



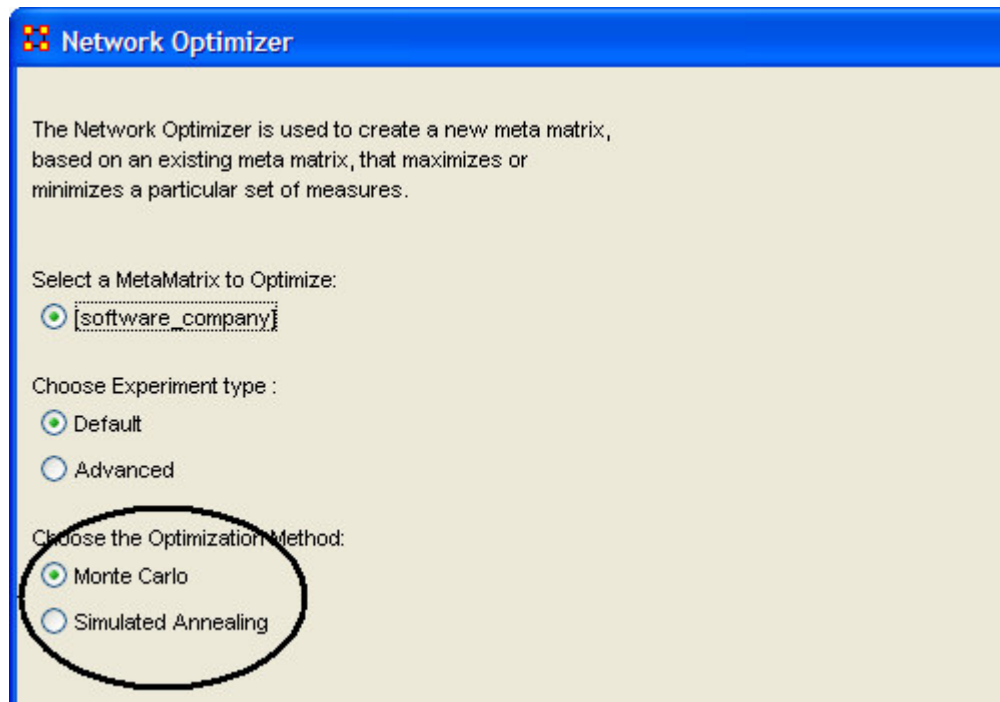
[Related topic "Default / Advanced" Experiment Type](#)

Optimization Method

Choose *Monte Carlo* for a totally random network optimization based on a variety of variables. *Simulated Annealing* allows for a more structured network optimization.

If you select *Monte Carlo* simply click next. *Simulated Annealing* requires you to select certain variables that will run your network optimization in a more structured manner.

The black ellipse below shows where to make this selection.



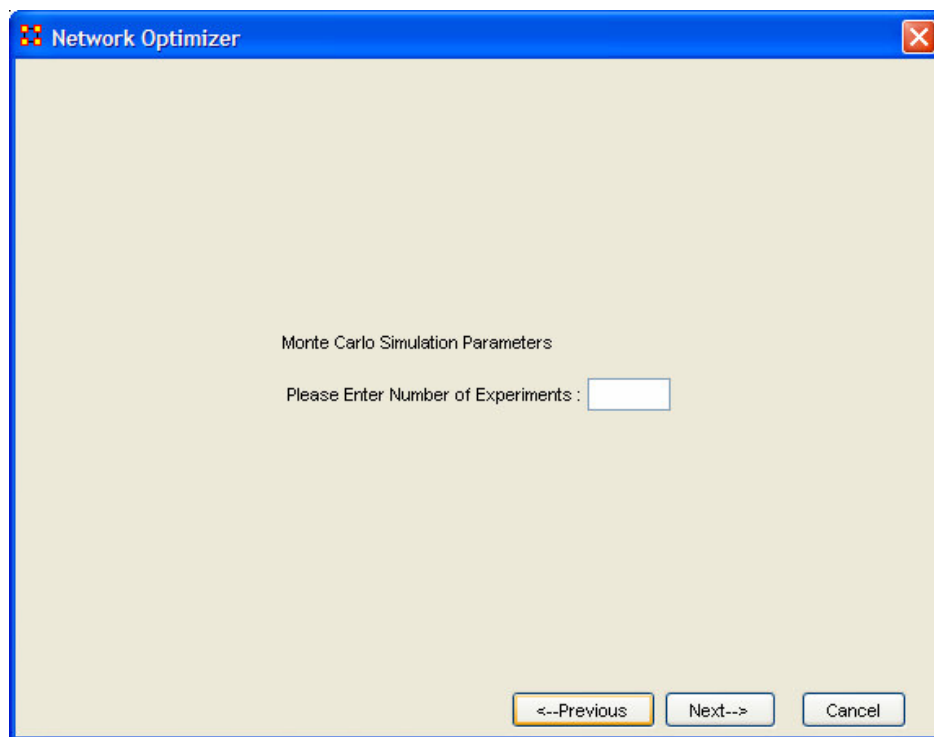
[Related topic "Monte Carlo" Optimization Method](#)

[Related topic "Simulated Annealing" Optimization Method](#)

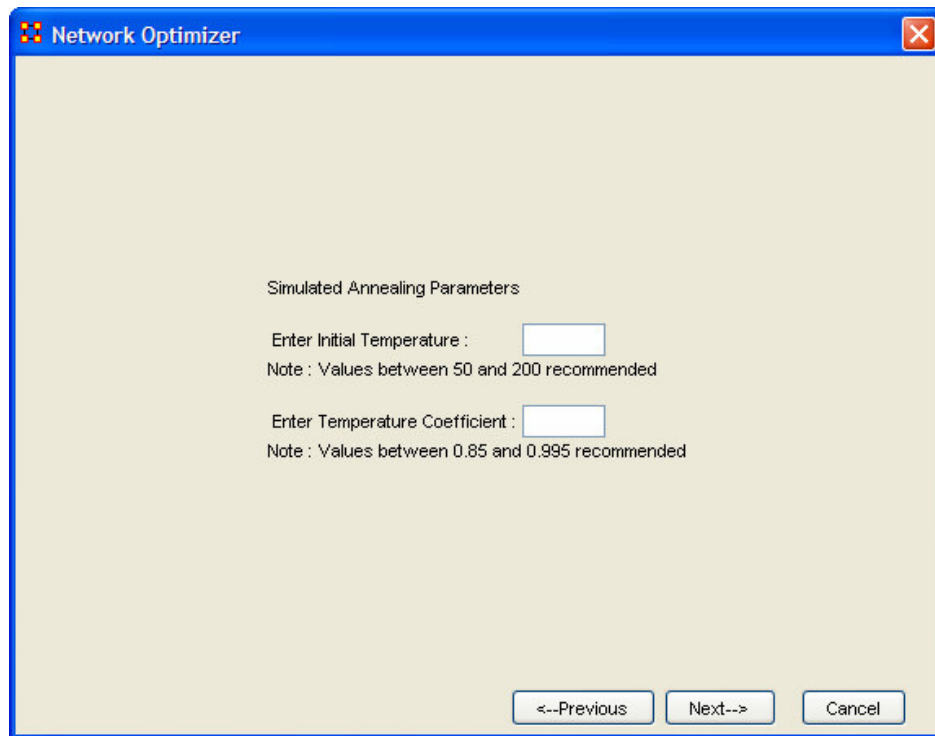
Default and Advanced Experiment Type

The Default and Advanced Experiment Type options are relevant to both the Monte Carlo and Simulated Annealing Optimization methods. If you select "Default" option in the Network Optimizer window pane, for either Monte Carlo or Simulated Annealing, no further input is necessary. Simply click the next button. If you choose "Advanced" the following screen appears (screen shot below) requesting parameter input respective to either the Monte Carlo or Simulated Annealing optimization method.

The Network Optimizer screen shot below requests parameter input for an "Advanced" Monte Carlo Simulation. The recommended value for the Monte Carlo Experiment is between 50,000 and 500,000. Following this screen shot is another, displaying the parameter input for the Advanced Simulated Annealing optimization.



Advanced Simulated Annealing Optimization Method:



"Initial Temperature" recommended value is around 100.

"Temperature Coefficient" recommended value is around 0.95.

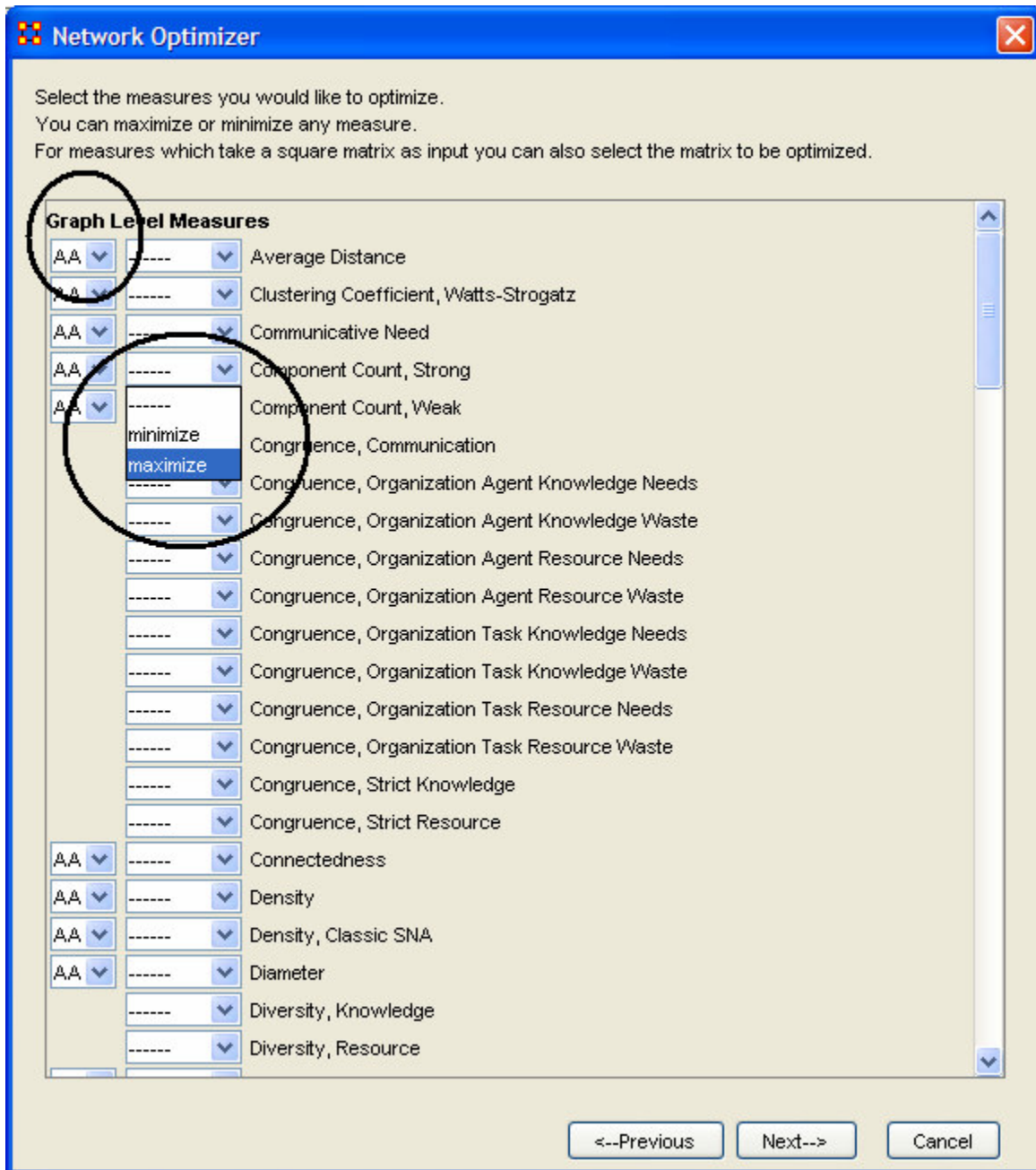
Tip! The term "Simulated Annealing" draws its inspiration from metallurgy, where essentially atoms within a metal are heated thereby dislodging them from a metal's internal structure transforming the metal into another atomic state. In this way, your organization is "heated" changing its components in the attempt to arrive at an optimized state.

[\(Related topic "Monte Carlo" Optimization Method\)](#)

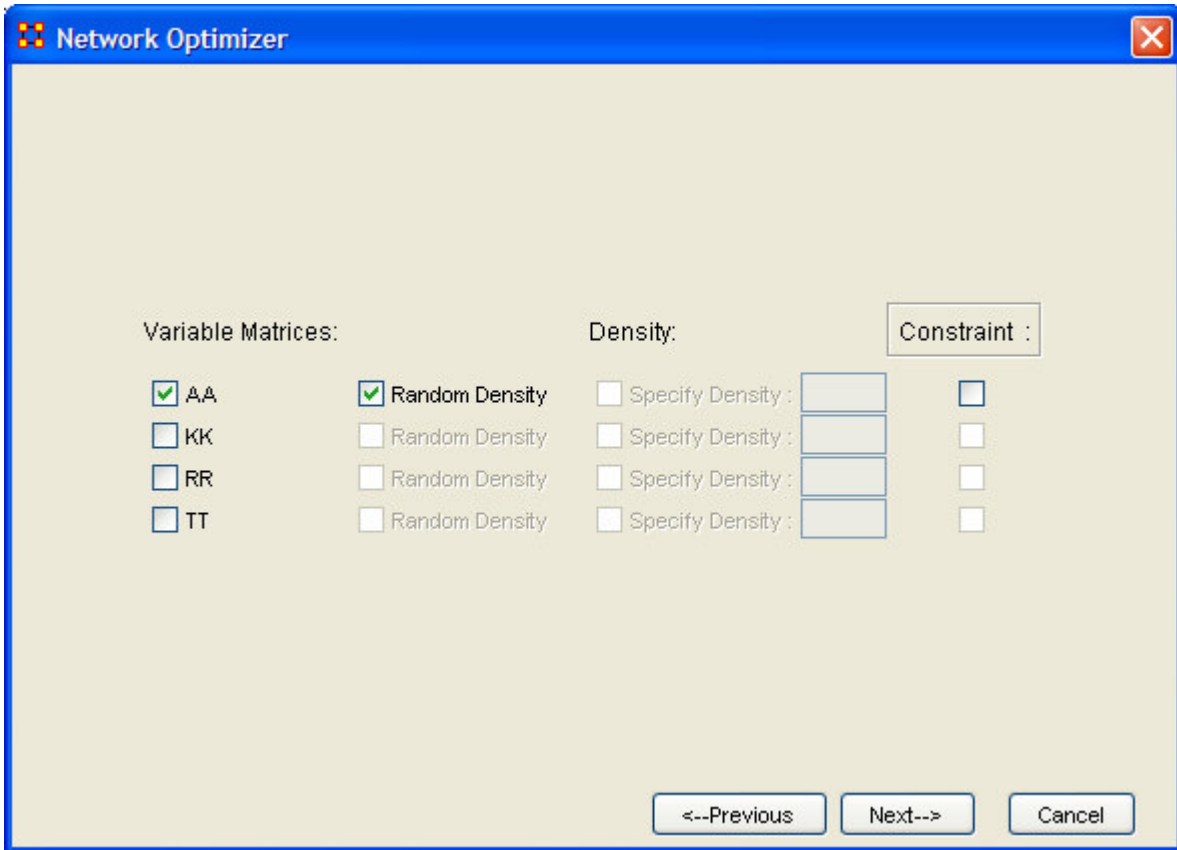
[\(Related topic "Simulated Annealing Optimization Method\)](#)

Monte Carlo Network Optimization

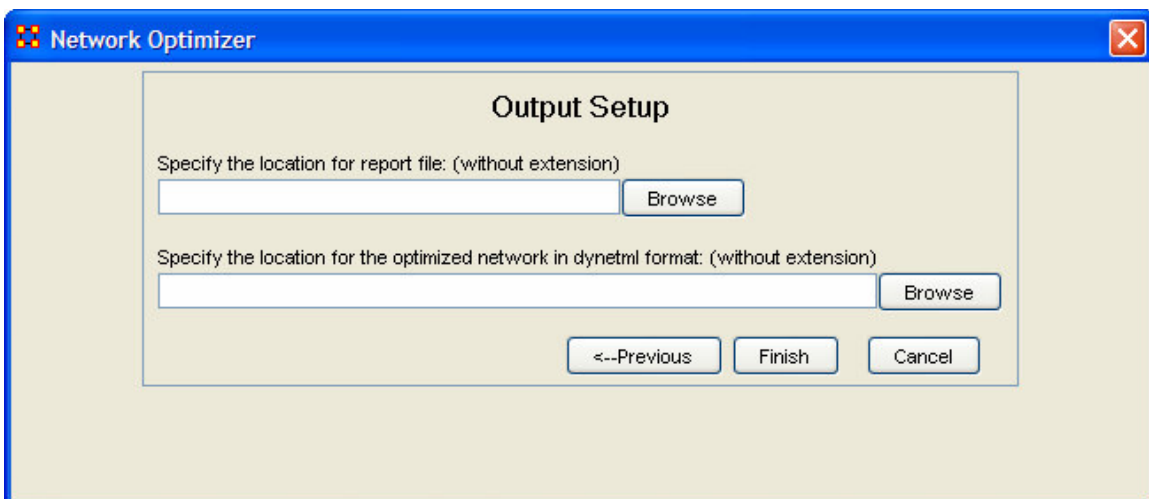
Monte Carlo, as the name implies, is a random optimization of your organization. To initiate a Monte Carlo Network Optimization, first select the measures you would like to optimize. Note that you can select multiple measures to be optimized simultaneously. Graph Level allows you specify the node type: *Agent, Agent; knowledge, knowledge; Task, Task; Resource, Resource*. Black ellipses below highlight areas of the Network Optimizer window pane that enable you to customize your network optimization.



After you have selected the measures you are interested in optimizing, click the "Next" button. Choose the sub-matrices to be varied during the optimization process. Please note: In the Monte Carlo method you can specify if you would like to run optimization with fixed or random density, and if you want to keep at least one non-zero element in every row of sub-matrices.



Click "Next." Specify the location for the data log file, without specifying an extension. Select "Verbose" if you want to analyze the whole process of optimization.



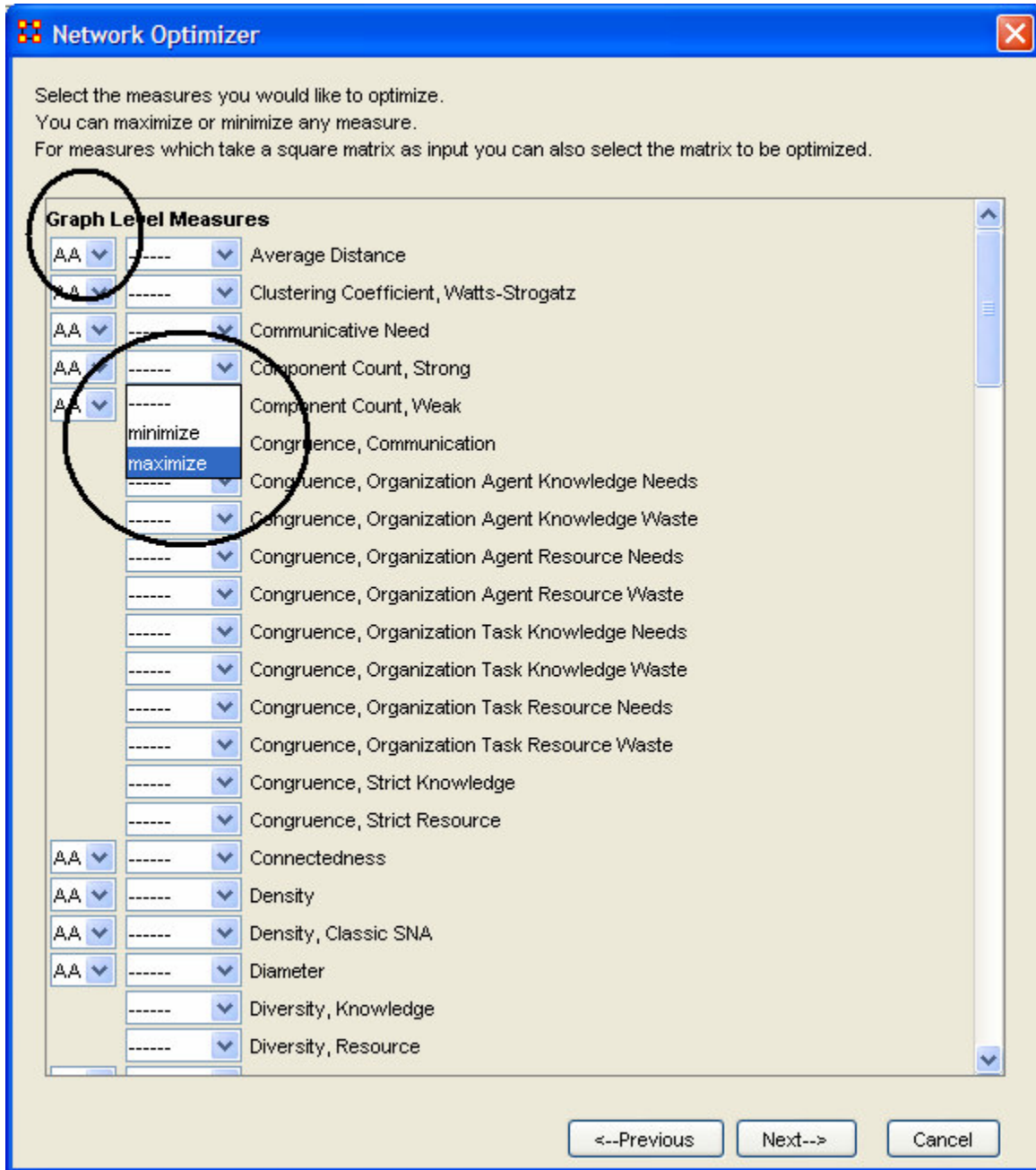
Click "Next." Specify network location(s) for the MetaMatrix output. This can be either in one MetaMatrix xml file or separate files for each Submatrix. In either case, specify the file location(s) without extensions (file extensions will be created by the optimizer).

Click "Next'" Check the "Add Organization to Meta Matrix Manager" box if you want to run ORA reports on the resulting Meta Matrix. Click "Generate" to start the optimization." The data log file will be displayed in the bottom panel of the main ORA Main Interface window.

[\(Related topic "Simulated Annealing"\)](#)

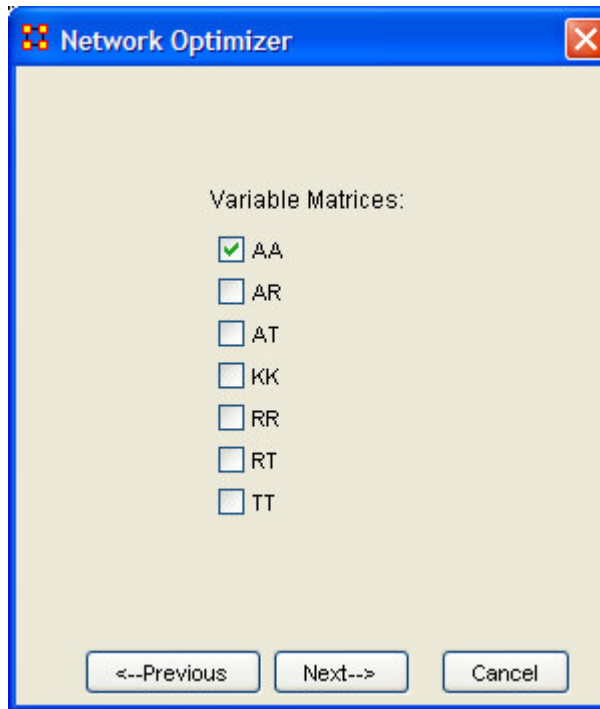
Simulated Annealing Network Optimization

Select the measures you would like to optimize. Note that you can select multiple measures to be optimized simultaneously. Graph Level allows you specify the node type (Agent, Agent; knowledge, knowledge; Task, Task; Resource, Resource). The black circles below highlight the areas of the Network Optimizer window pane that allow you to customize your network optimization.

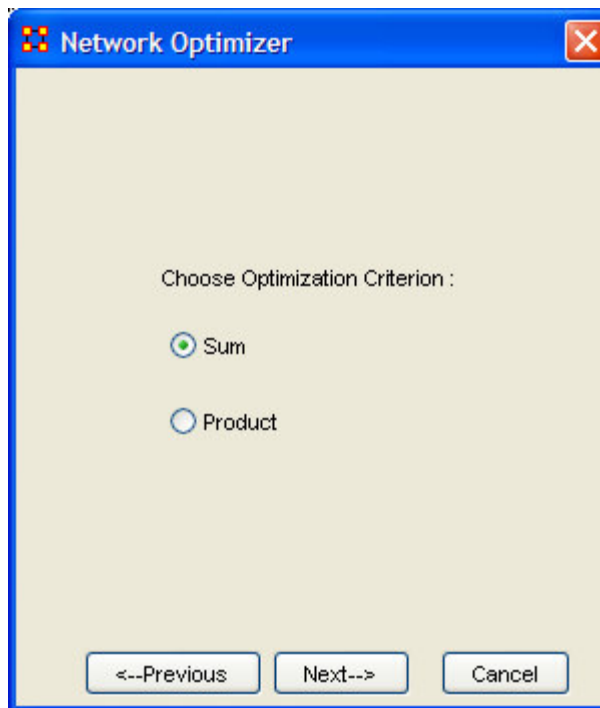


After you have selected the measures you are interested in optimizing, click the "Next" button. Choose the sub-matrices that to be varied during the optimization process. If you

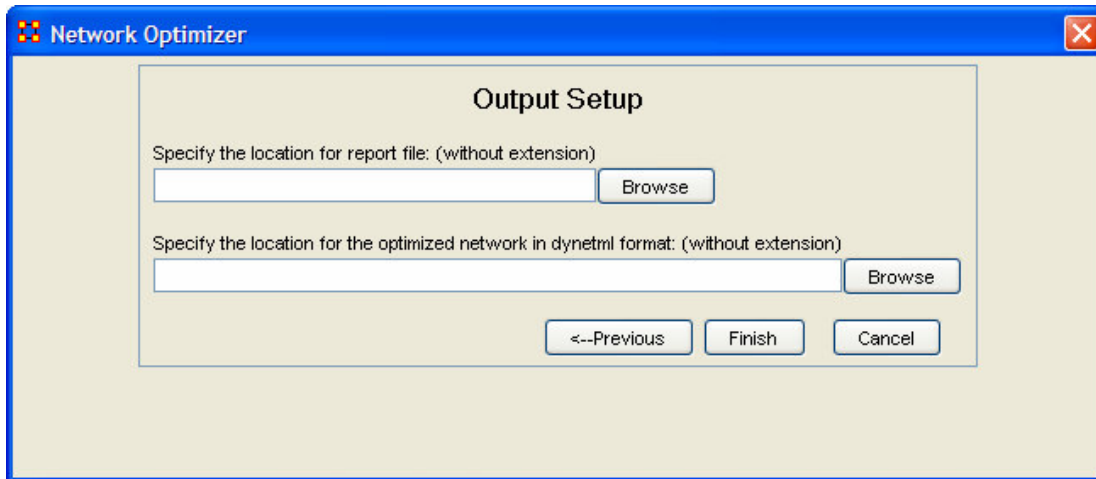
chose Monte Carlo method you should also specify if you would like to run optimization with fixed or random density, and if you want to keep at least one non-zero element in every row of sub-matrices.



Click "Next." Then choose the optimization criterion: Sum or Product.



Click "Next." Specify the location for the data log file, without specifying an extension. Select "Verbose" if you want to analyze the whole process of optimization.



Click "Next." Specify network location(s) for the MetaMatrix output. This can be either in ONE MetaMatrix xml file or separate files for each submatrix. In either case, specify the file location(s) without extensions (file extensions will be created by the optimizer).

Click "Next" Check the "Add Organization to Meta Matrix Manager" box if you want to run ORA reports on the resulting Meta Matrix. Click "Generate" to start the optimization." The data log file will be displayed in the bottom panel of the main ORA Main Interface window.

[\(Related topic "Monte Carlo"\)](#)

Over-Time Viewer

The Over-Time Viewer enables you to study changes within your organization or network over a time period. For instance, the overall "centrality" value of your network can be analyzed as it relates to network data compiled over the years, say, 2000, 2001, and 2002. The time interval is dependent only on your data collection samples. Such an analysis can then be compared to external or internal events.

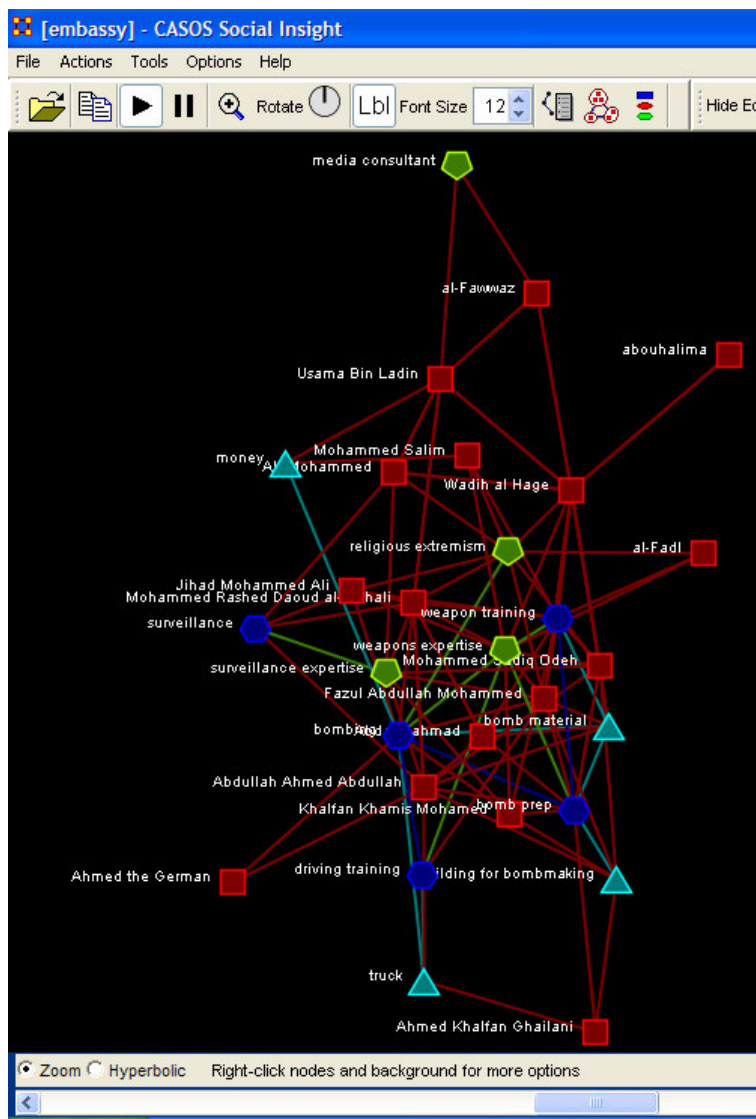
As an example, let us say you are interested in learning how the events of September 11, 2001 affected a terrorist organization or how the passing of anti-terrorism legislation impacted the same network. In either case, you can run measures in the Over Time Viewer on your network samples (loaded as multiple MetaMatrices) then compare the results against such external events.

[Running An Over-Time Analysis](#)

The ORA Visualizer

The ORA Visualizer renders a MetaMatrix graphically. You can interact with your data in a variety of ways: remove key actors, isolate certain links, or focus on any particular relationship by using tools such as the Path Finder and grouping algorithms.

Below is a visualized MetaMatrix of the al Qaeda terrorist network believed to be behind the 1998 U.S. Embassy bombing in Dar es Salaam, Tanzania. In this example, red squares denote actors; light blue triangles: resources; green pentagons: knowledge bases; blue pentagons: tasks. The shapes are called *nodes*. The colored lines, which link nodes together, represent a connection or direct relationship to each other. The terms *edge*, *tie*, and *link* are used interchangeably to describe these connections.



[\(Simplifying A Complex Visual Network\)](#)

Visualizer Tools

The ORA Visualizer provides a suite of tools to visually analyze your MetaMatrix:

[Drill Down](#)

[Entity Status](#)

Group Viewer

[Key Set Selector](#)

[Meta-Nodes](#)

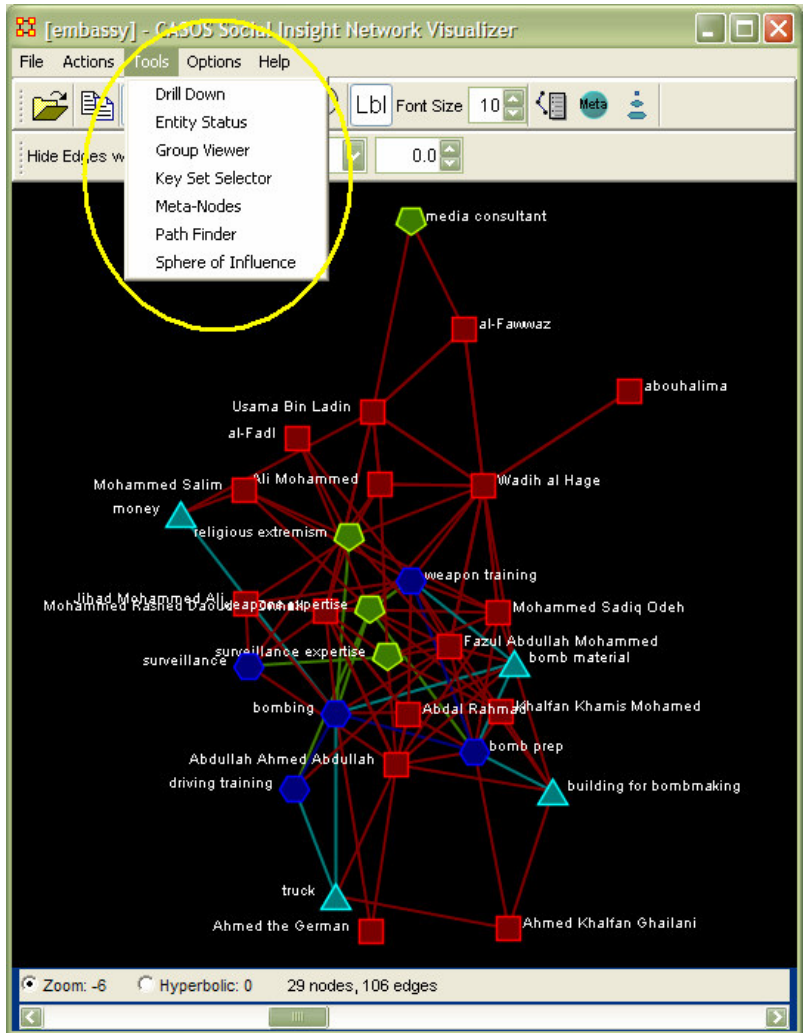
[Path Finder](#)

[Sphere of Influence](#)

To access these tools, you need to be in the Visualizer:

[ORA Visualizer drop down menu > Tools](#)

The yellow ellipse in the screen shot below highlights where to access the Visualizer Tools menu.



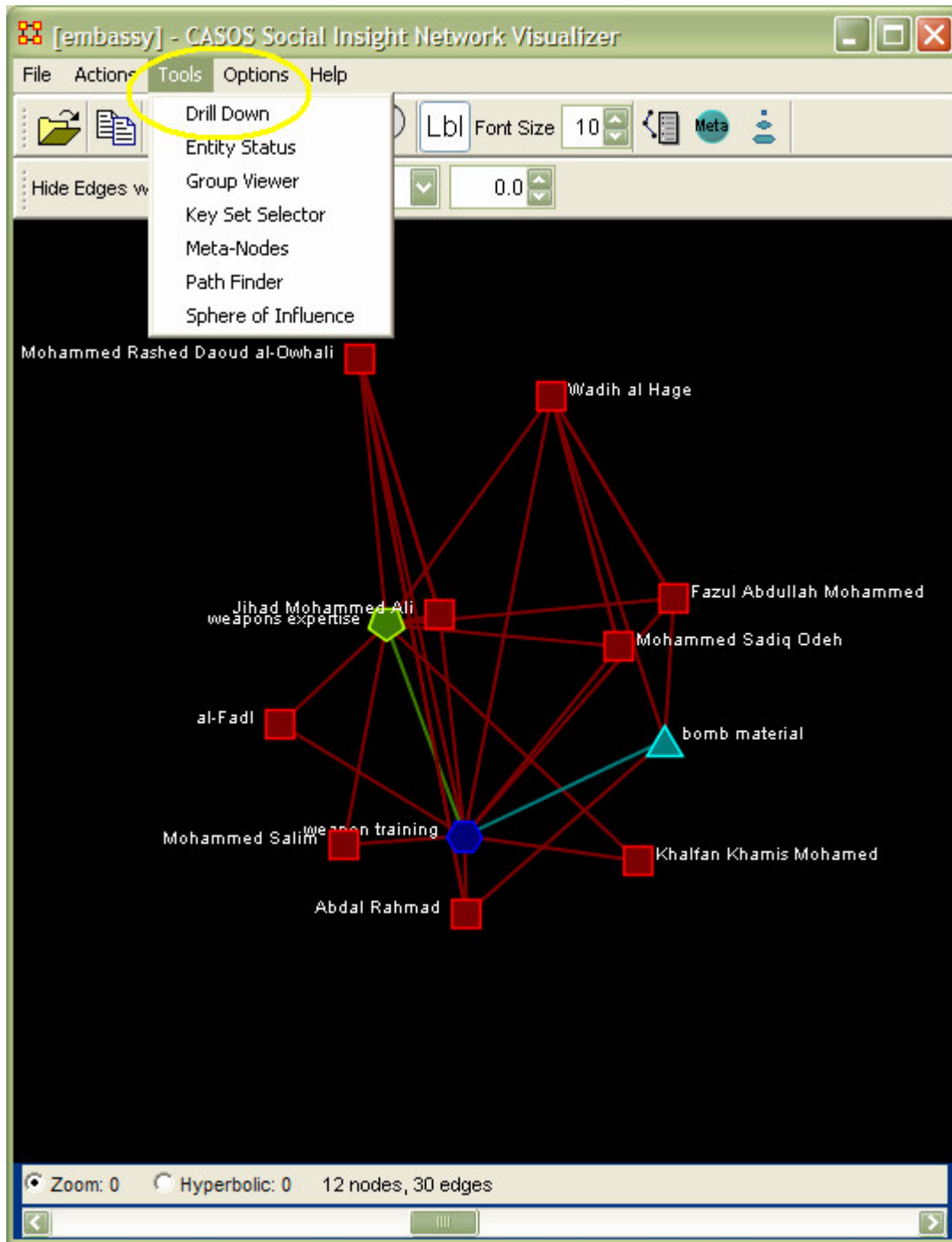
[Back to Visualizer](#)

Drill Down Wizard

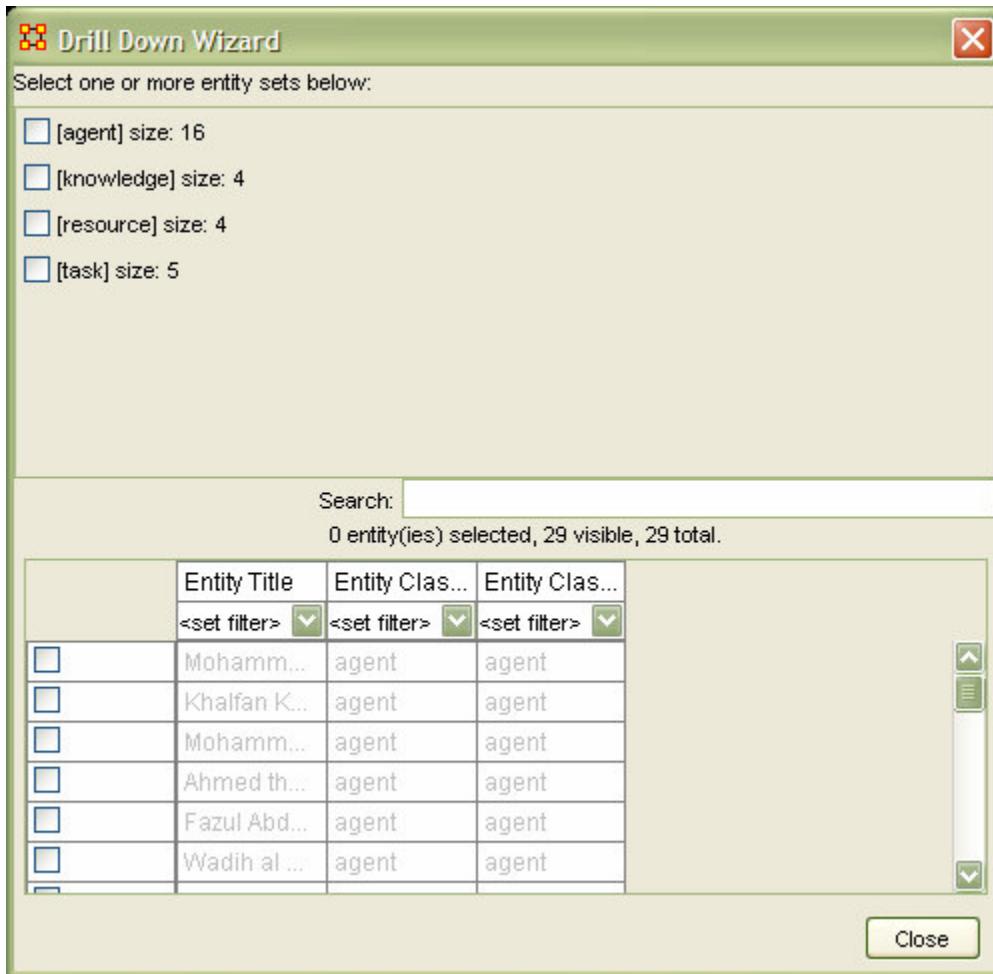
With the ORA Drill-Down Wizard you can quickly visualize ego networks by overall entity class (e.g. knowledge, tasks, resources, agents) or by choosing individual nodes from a checklist. The ORA Drill-Down Wizard is only accessible through the Visualizer:

[From the ORA Visualizer drop down menu > Tools > Drill Down](#)

The ellipses in the screen shot below highlight how to access the Drill-Down Wizard from the Visualizer:



Select Drill-Down from the drop down menu and the following window box should appear (screen shot below).



[Drill-Down Wizard Explained](#)

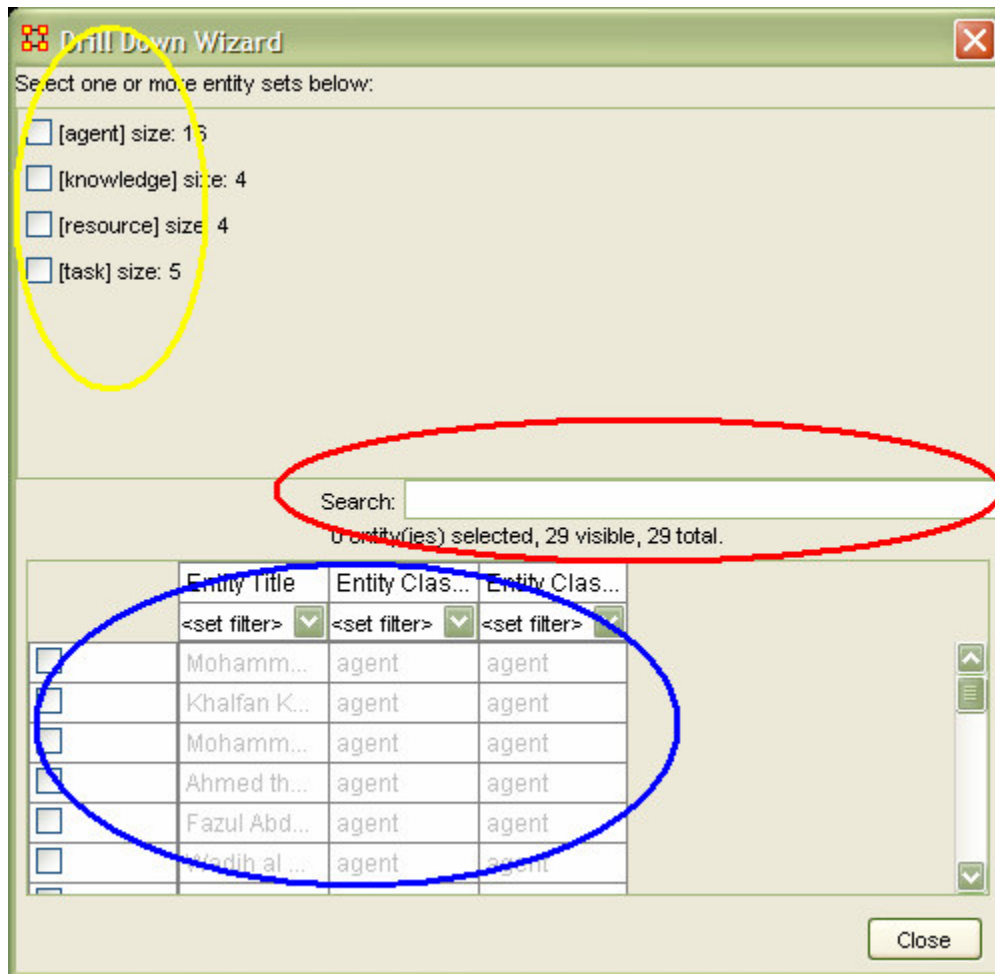
[Drill-Down Wizard Example](#)

[Back to Visualizer](#)

Drill-Down Wizard Explained

The ORA Drill-Down Wizard can be broken down into three primary sections of input:

- 1) The first section (yellow ellipse below) enables you choose individual entity sets or combination of entity sets to display in the Visualizer.
- 2) The second section (red ellipse below) of the Drill-Down Wizard enables you to search for a particular node within your MetaMatix. This can be handy when you dealing with large nodesets and the one you are interested in finding is not easily located.
- 3) The final section (blue ellipse below), enables you "check mark" an individual node within your MetaMatrix by entity class if so desired.



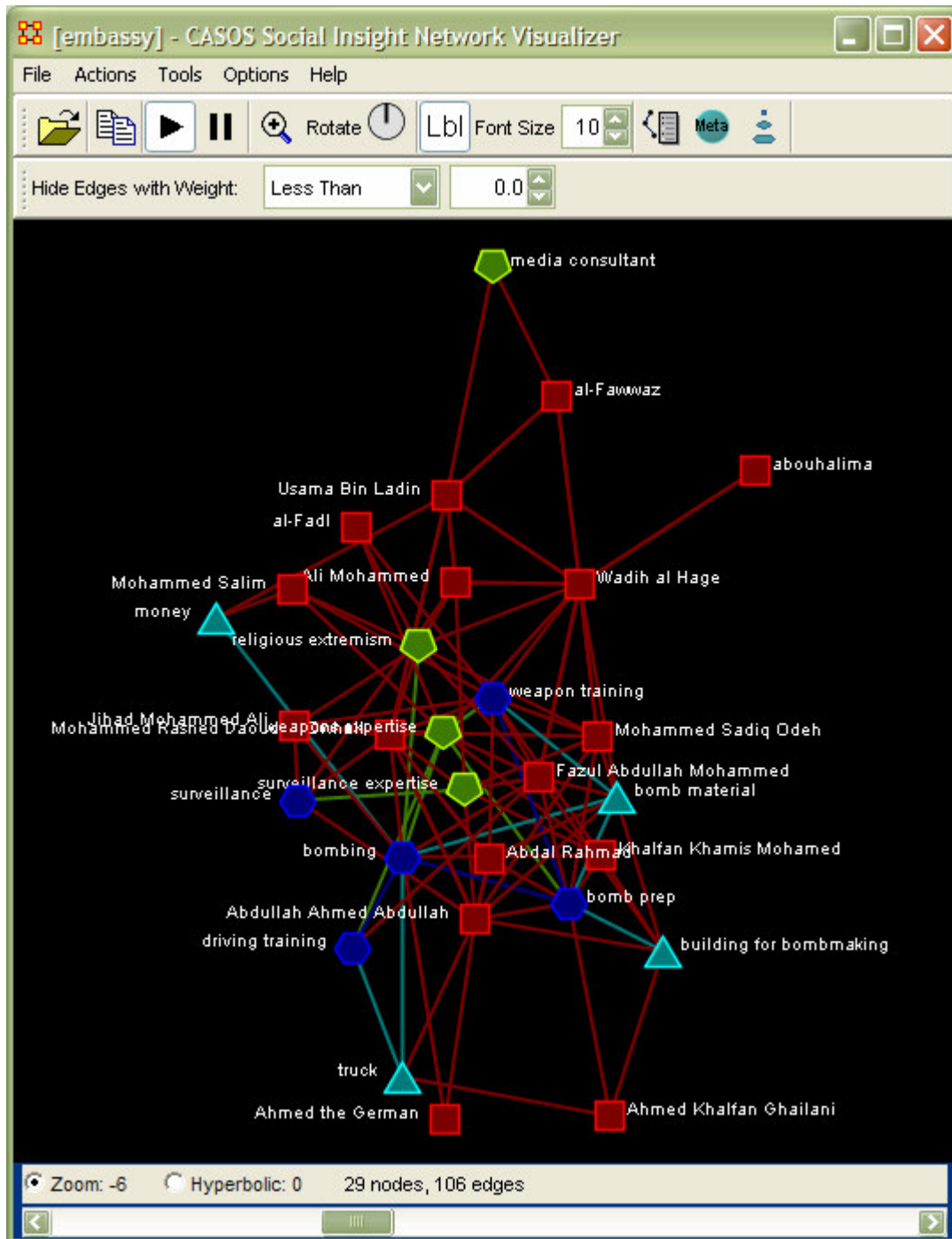
Back to Drill-Down Wizard

Drill-Down Example

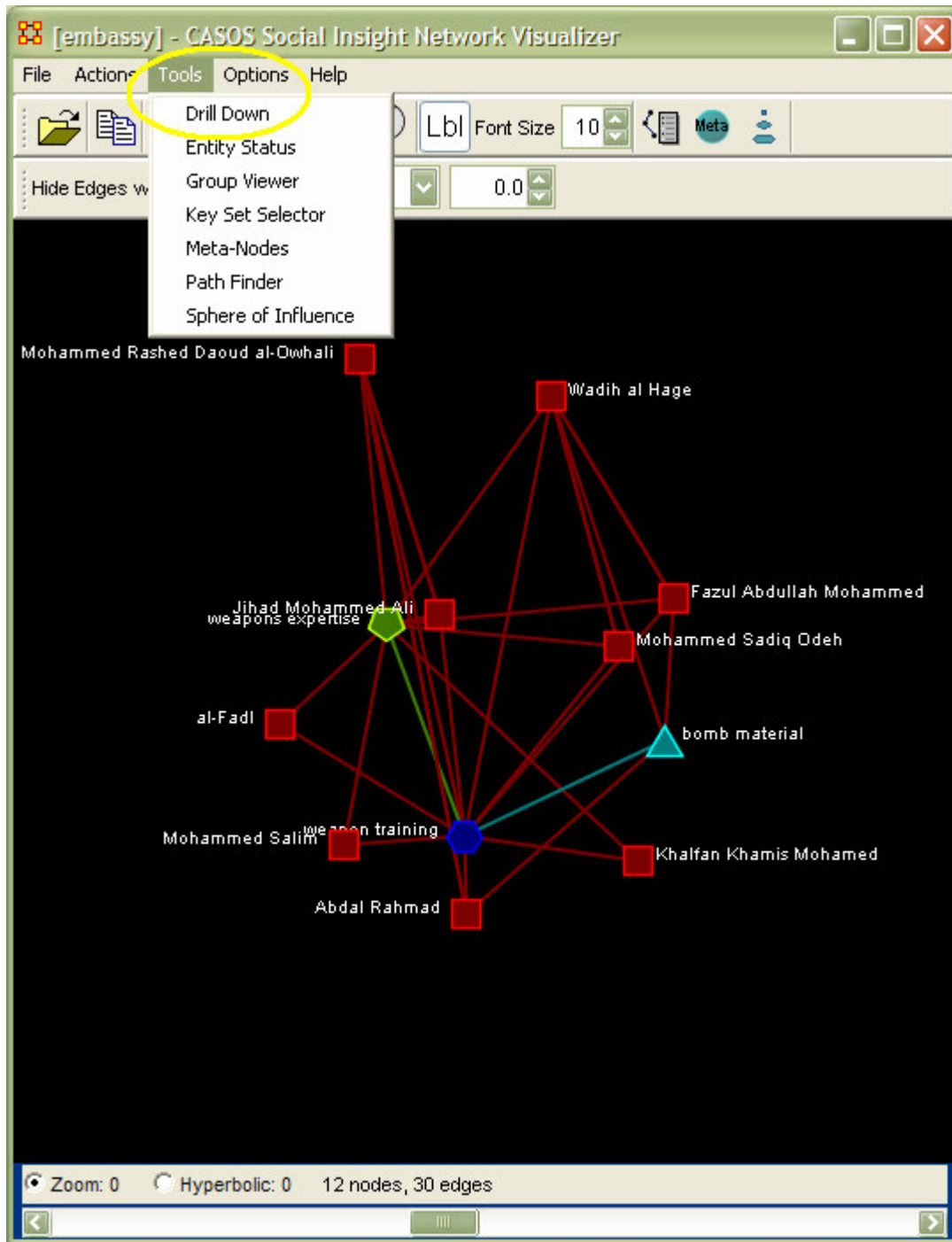
[Back to Tools](#)

Drill Down Wizard Example

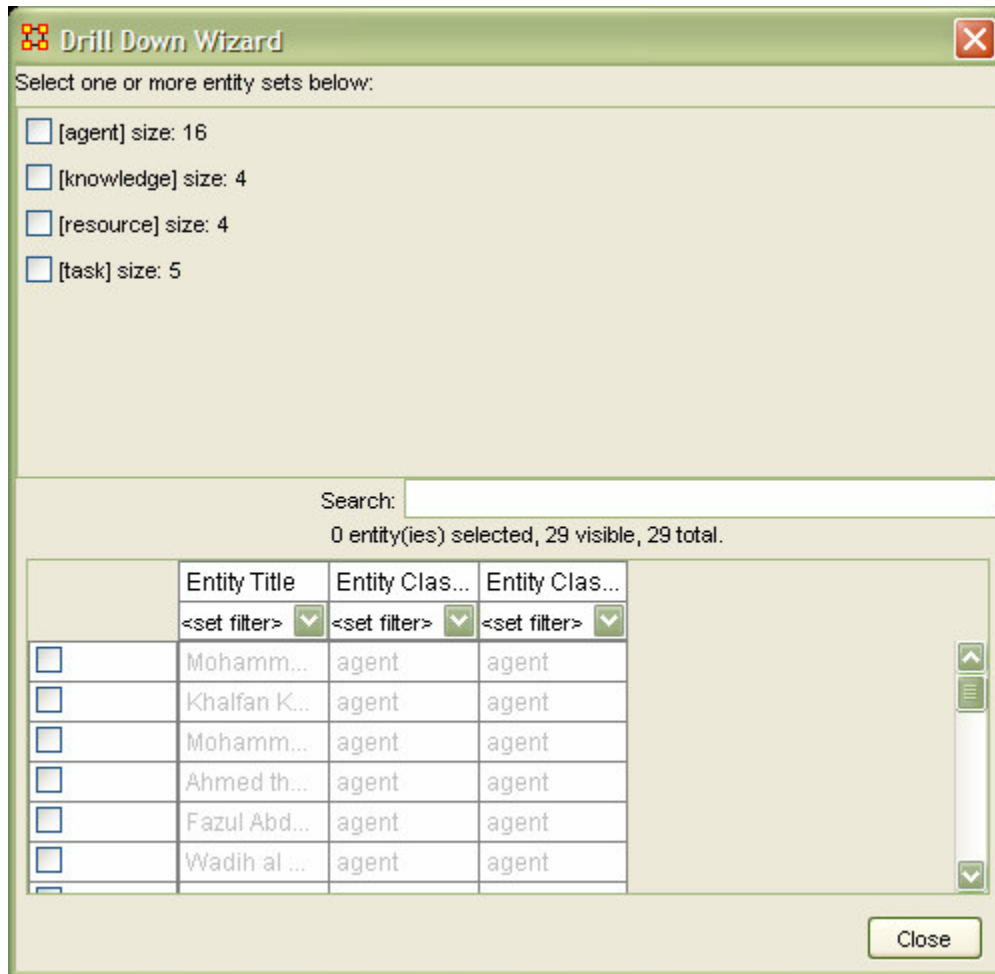
Using the Embassy MetaMatrix, we will render the various entity classes using the Drill-Down Wizard Tool. The screen shot below displays the Embassy MetaMatrix as it should first appear in the Visualizer.



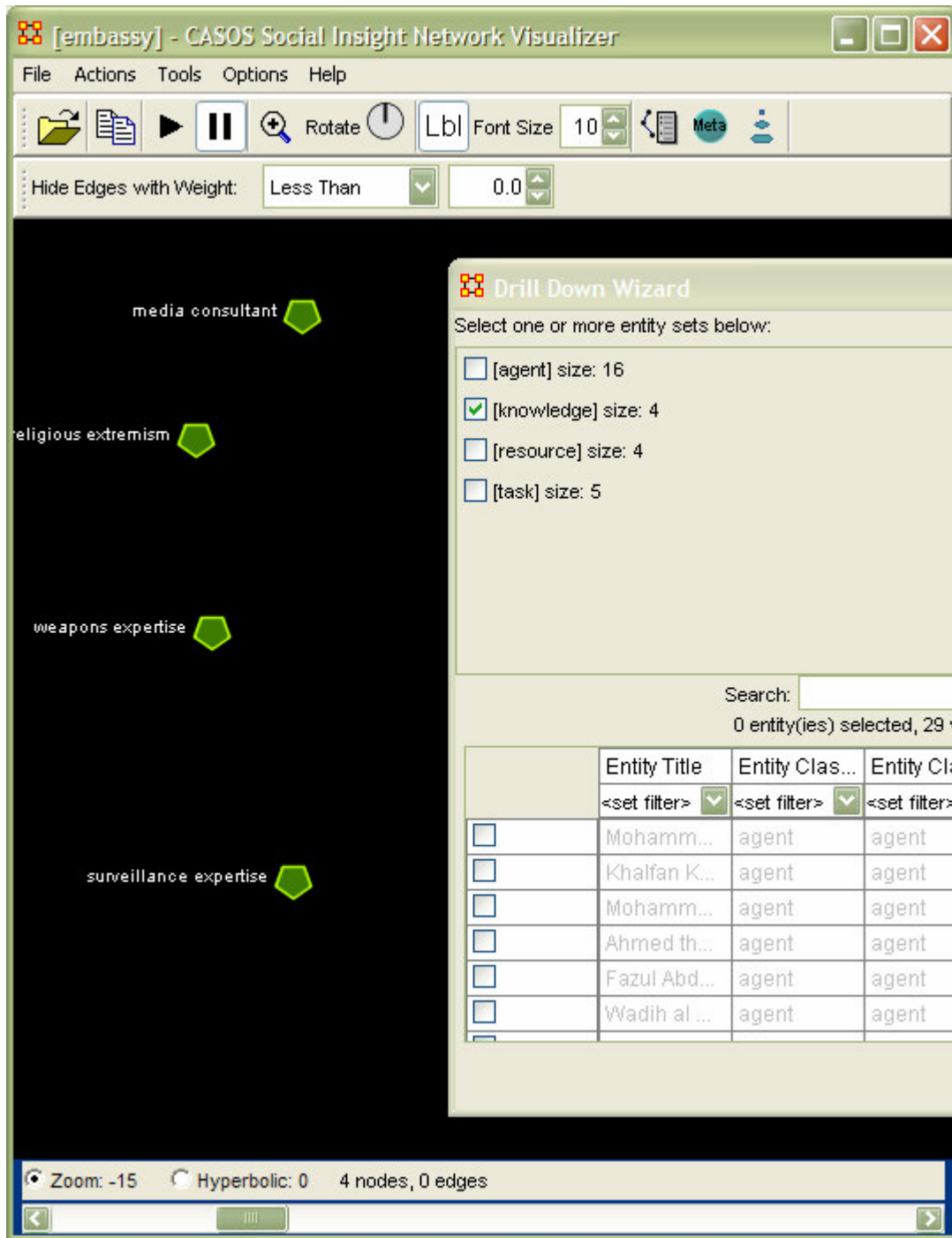
Next, select the Drill-Down Wizard from the Visualizer Tool Bar (screen shot below).



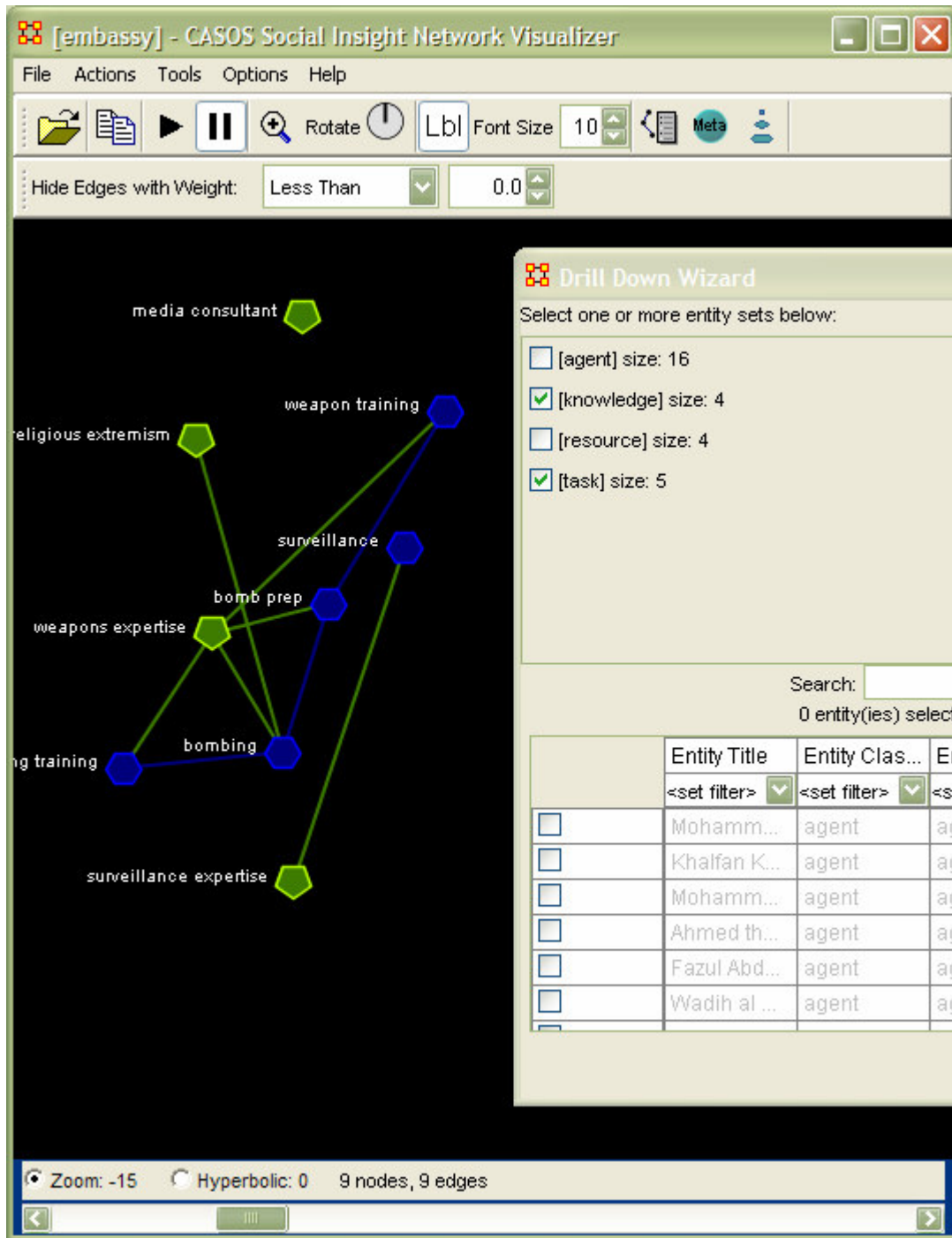
The Drill-Down Wizard should appear (screen shot below).



Here, select the entity class Knowledge by selecting the Knowledge check mark box. The screen shot below, displays the end result that you should now see in the Visualizer.



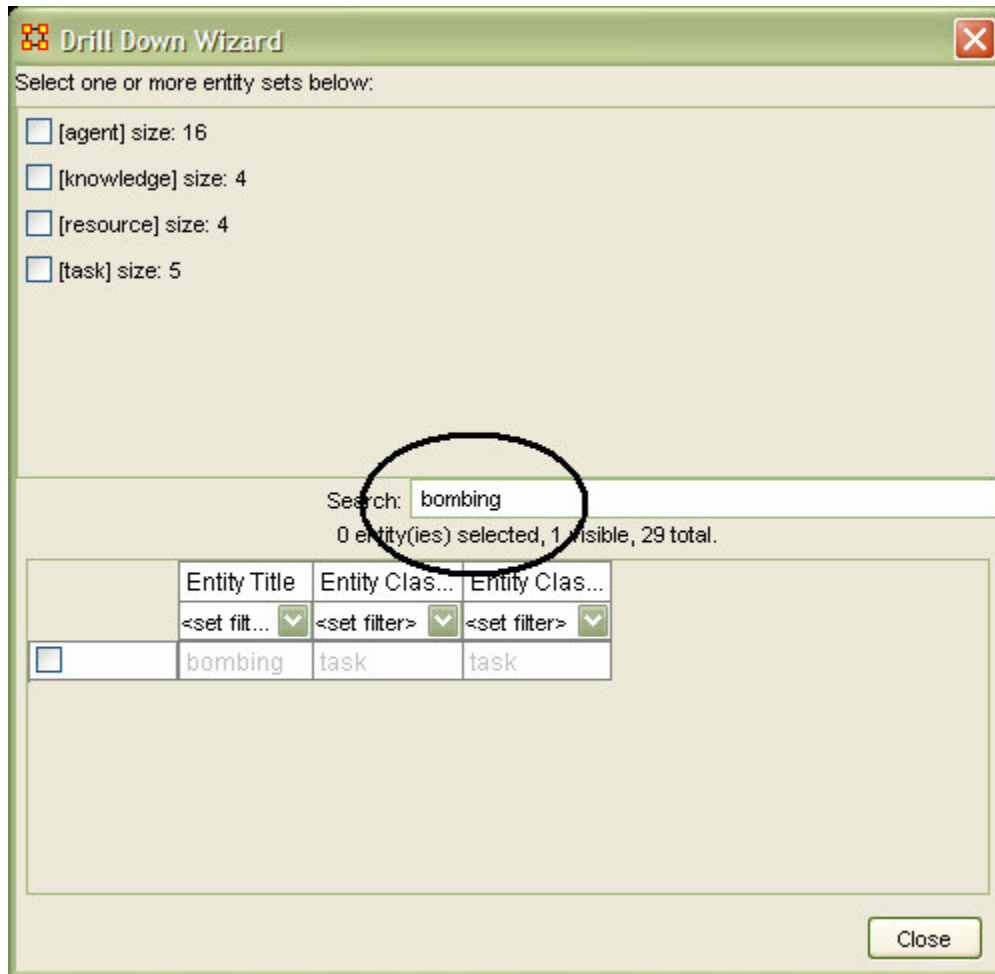
Next, with the Knowledge entity box checked, add the entity set Task. The screen shot below displays the end result.



Now we will use the search bar feature of the Drill-Down Wizard.

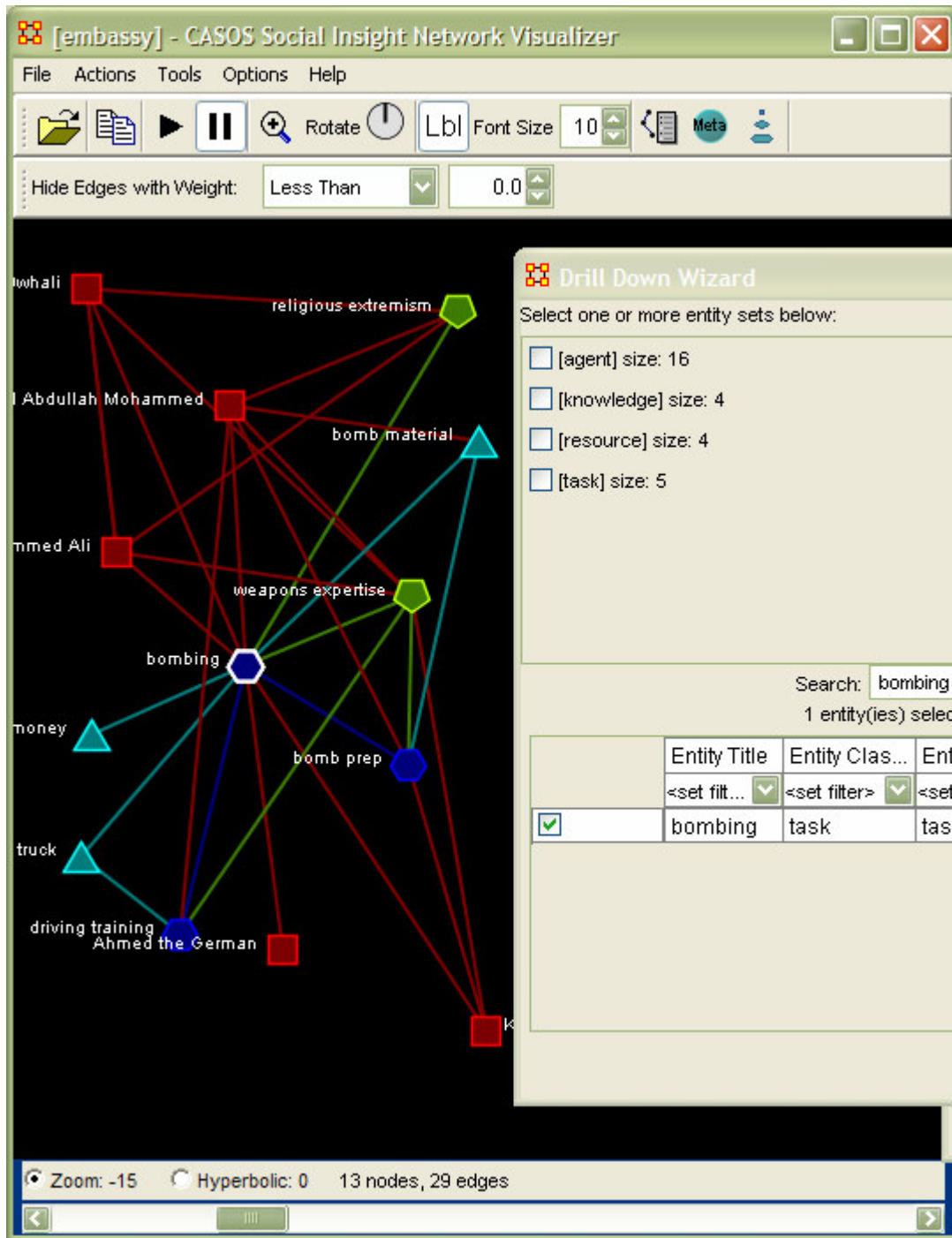
To begin, de-select all the entity class check boxes so that nothing appears in the Visualizer.

Next, enter the word "Bombing" in the Visualizer search field below (screen shot below)



In the above screen shot, you should see "bombing" has been isolated from our Embassy Visualization and is the only node displayed in the bottom section of the Drill-Down Wizard.

Next, select the bombing check box. The ego map for task entity "bombing" should now be displayed (screen shot below).



[Back to Drill Down Wizard](#)

[Back to Drill Down Wizard Explained](#)

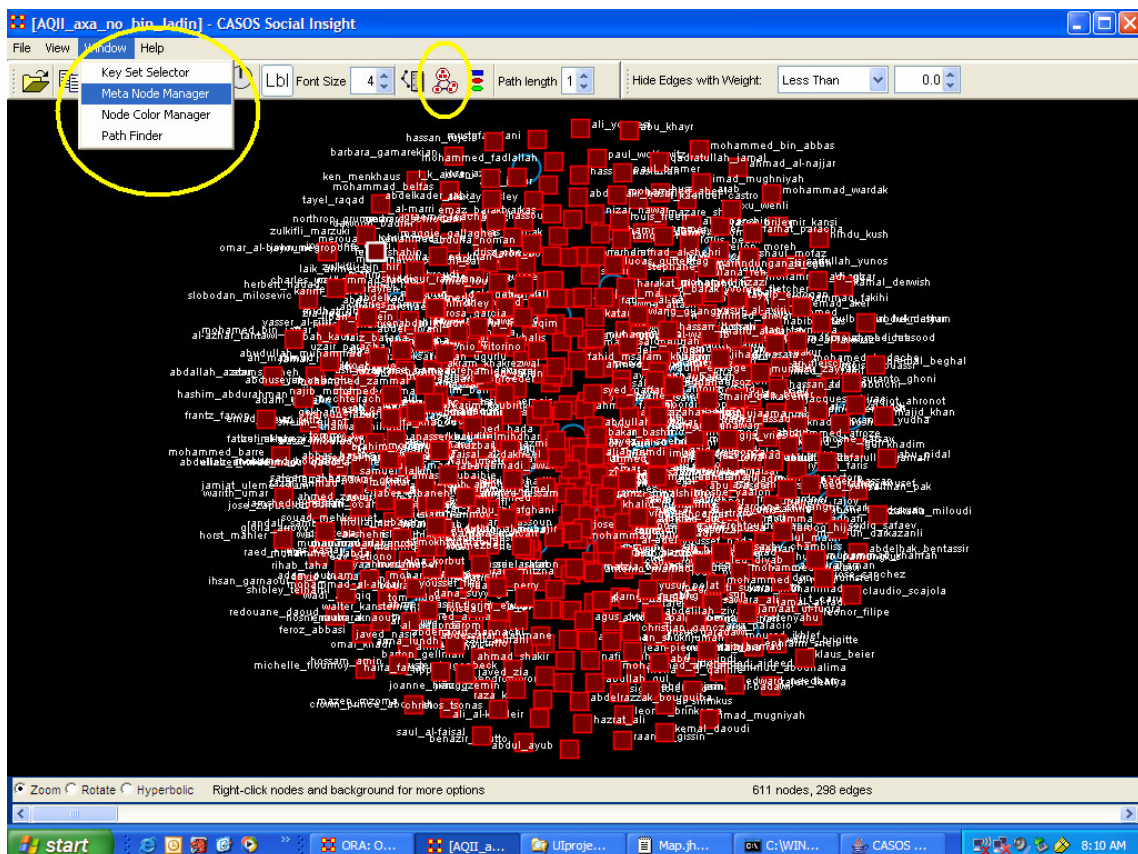
[Back to Tools](#)

Creating A MetaNode

A *MetaNode* contains multiple nodes collapsed into one. You can create MetaNodes based on the nodesets in your organization, or you can create MetaNodes based on the attributes of the nodes. To create MetaNodes, you must access the *MetaNode Manager*. There are two ways to do this task:

1. From the drop down menu > Window > MetaNode Manager
2. from the Visualizer tool bar

The yellow ellipses below highlight where to access the MetaNode Manager through the drop down menu and the Visualizer tool bar.



In the screen shot below, the yellow ellipses highlight how to create MetaNodes based on *Attribute 1* of our *Agent by Agent MetaMatrix* graph. The Visualizer itself shows your condensed visualization. All the nodes, which share the same attributes, are now groped into MetaNodes.

To view all the original nodes, click expand all MetaNodes. To create additional MetaNodes, click Create MetaNodes and select another attribute. You can only create MetaNodes based on defined attributes of your MetaMatrix.

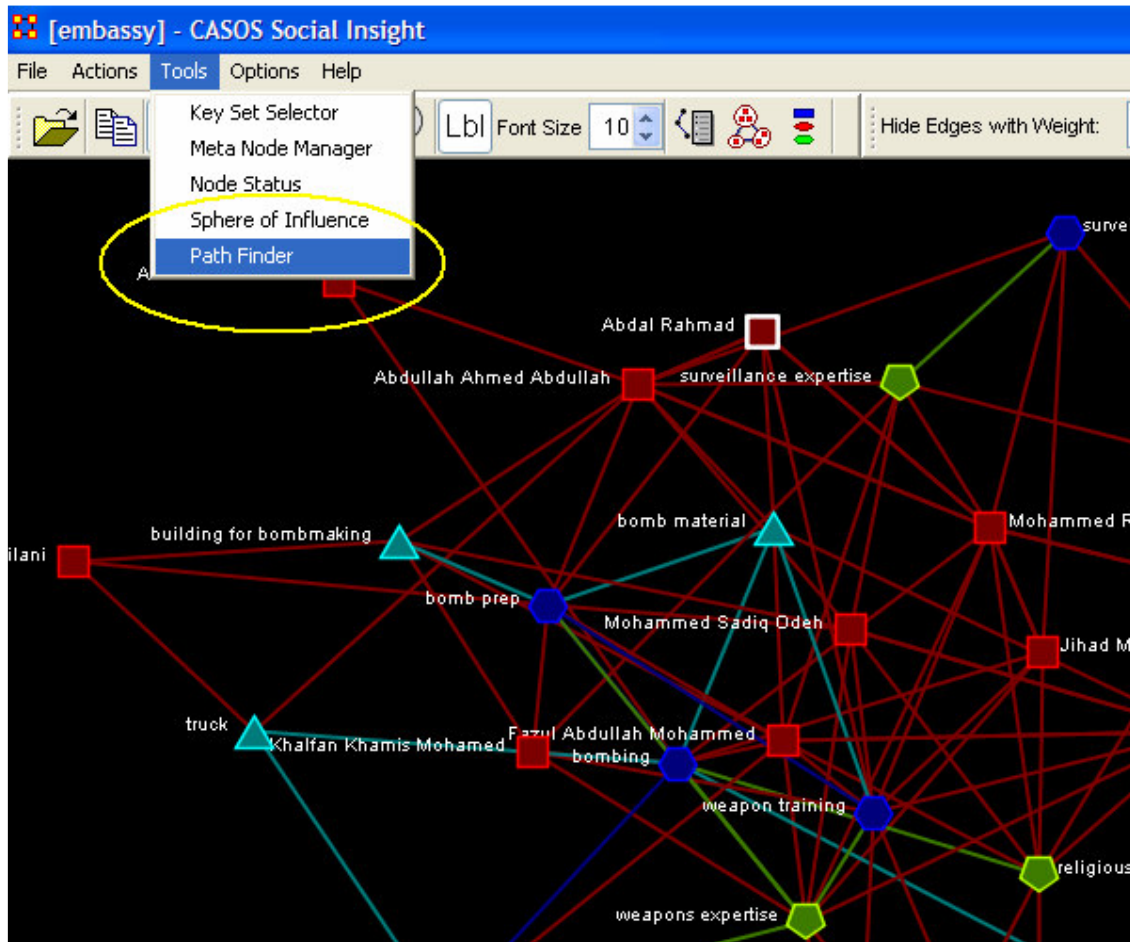
The Path Finder

The Path Finder allows you to focus or "drill down" on a particular node, or multiple nodes, that you may be interested in analyzing in greater detail. This is accomplished within the ORA Visualizer by using the Path Finder tool, accessible from the Visualizer tool bar. The Path Finder creates an "Ego Network" for any particular node or selection of nodes and can show you how nodes are connected.

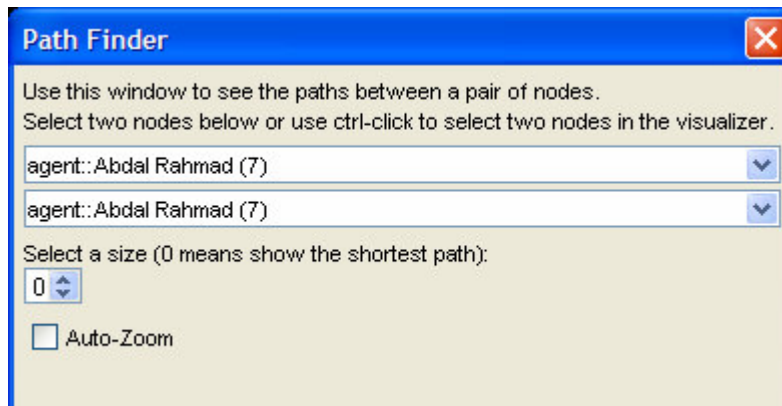
An Ego Network, or Sphere of Influence, is essentially a visual representation of a selected node and its relationship to its immediate neighbors, or other nodes, within the network. Each "direct tie" between a node and its neighboring nodes in a network is referred to as a "path." Path length is the number of ties that separate any two nodes. Multiple nodes can be used by the Path Finder for comparing Ego Networks.

The yellow ellipse in the screen shot below shows how to access the Path Finder tool from the ORA Visualizer.

From the drop down menu (in Visualizer) > Tools > Path Finder



After you select the Path Finder, the following Path Finder pop-up window appears:

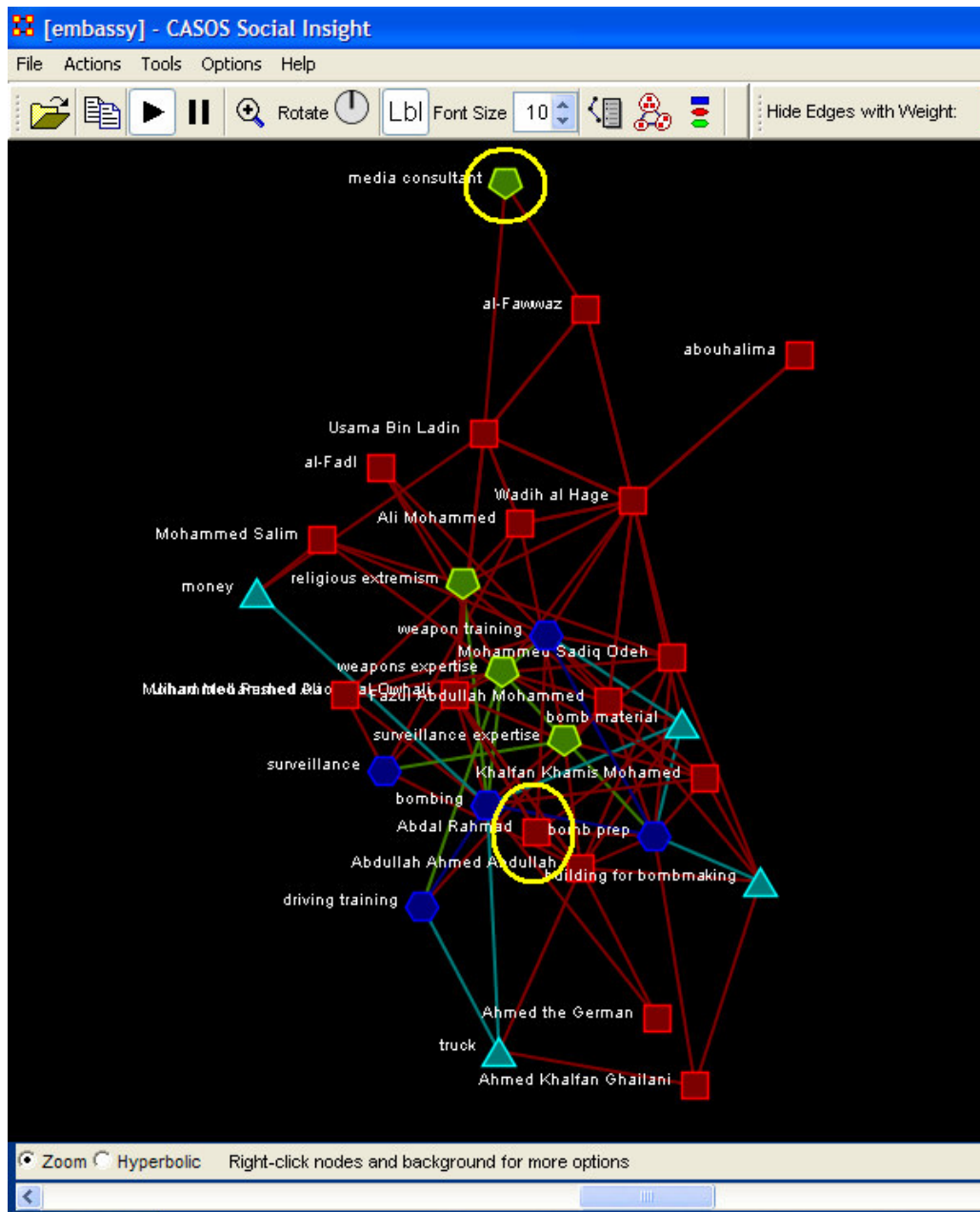


From the drop down selector bars, you can select particular nodes to visualize the "path" between them. This can also be accomplished by using control-click to select one or more nodes. Select a size defaults to "0." This shows the shortest path between two nodes. Increasing this number will increase the "path length" shown in the Visualizer. For instance, changing this to "1" will show paths between the nodes through neighbors in common. Check the Auto-Zoom box to automatically maximize your Path Finder visualization in the interface. In the example above, the agent node "Abdal Rahmad" is selected. By clicking the drop down arrow selector, this agent node can be compared to others within the network.

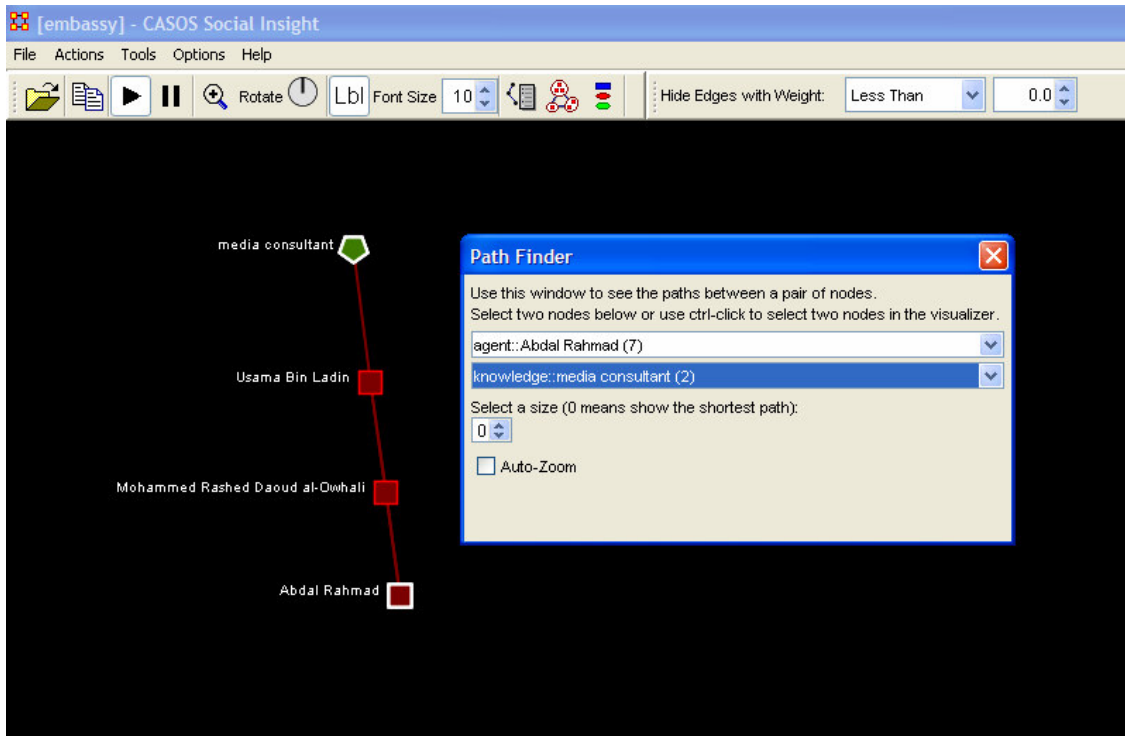
[Path Finder Example](#)

Path Finder Example

In this example, we are working with the MetaMatrix "Embassy." The screen shot below shows this MetaMatrix as it appears when first loaded in the Visualizer. We will attempt to find the "path" between the agent node "Abdal Rahmad" and the knowledge node "Media Consultant." The yellow ellipses below highlight the nodes whose paths we will attempt to visualize.



First, we access the Path Finder from the Tools menu.

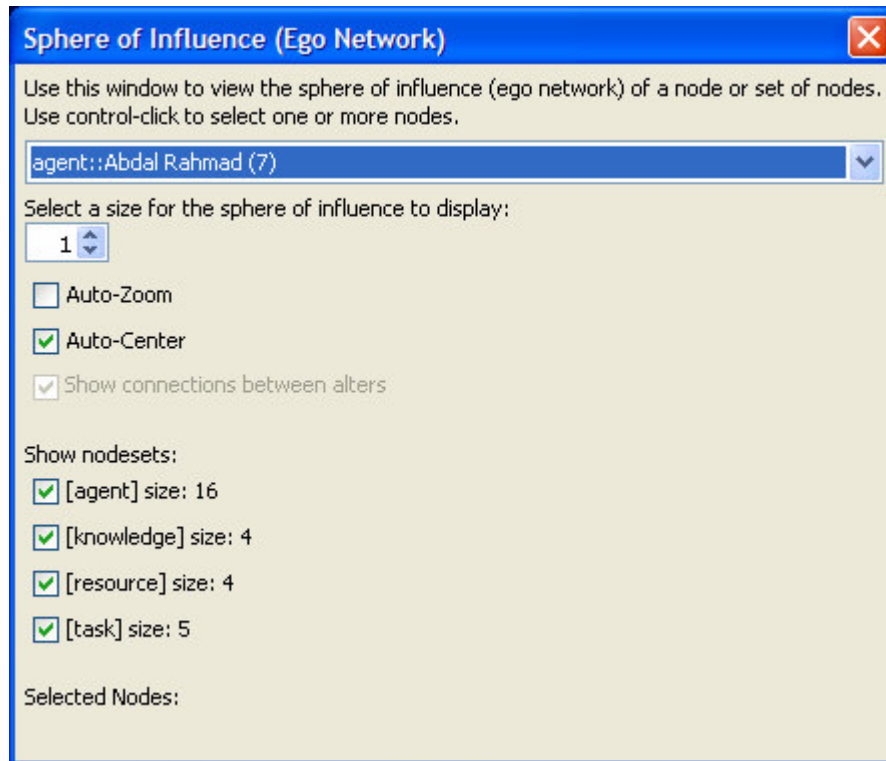


Sphere of Influence

Each node within a network has a unique "Sphere of Influence" or "Ego Network," essentially it's direct relationship with it's neighbors as a function of specified path length. The ORA Visualizer allows you to focus on this relationship by creating an "Ego Map" centered on any particular node you choose.

[From within the ORA Visualizer tool bar > Tools > Sphere of Influence](#)

The yellow ellipse below highlights where to access the Sphere of Influence tool from within the Visualizer tool bar. Scroll down for a link to a step-by-step example of using this tool.



Sphere of Influence Window

The **Drop-down Bar** arrow selector allows you to pick the node or set of nodes whose Sphere of Influence you are interested in obtaining. In the screen shot above, the first agent node "Abdal Rahmad" is selected.

Size Selector box allows you to choose the "path length" you are interested in for that particular node or set of nodes.

Auto Center will keep the visualized Ego Network centered within the Visualizer display.

Auto-Zoom will maximize your ego network within the Visualizer.

Show Nodesets allows you to select the components you wish to display in a node's Sphere of Influence. For instance, in the example above if you were only interested in the the "resources" directly tied to the agent node "Abdal Rahmad" then you would de-select (or un-check) the boxes corresponding to the nodesets agent, knowledge and tasks leaving only resources checked. This will produce a resource ego map for the agent node "Abdal Rahmad."

Sphere of Influence Example

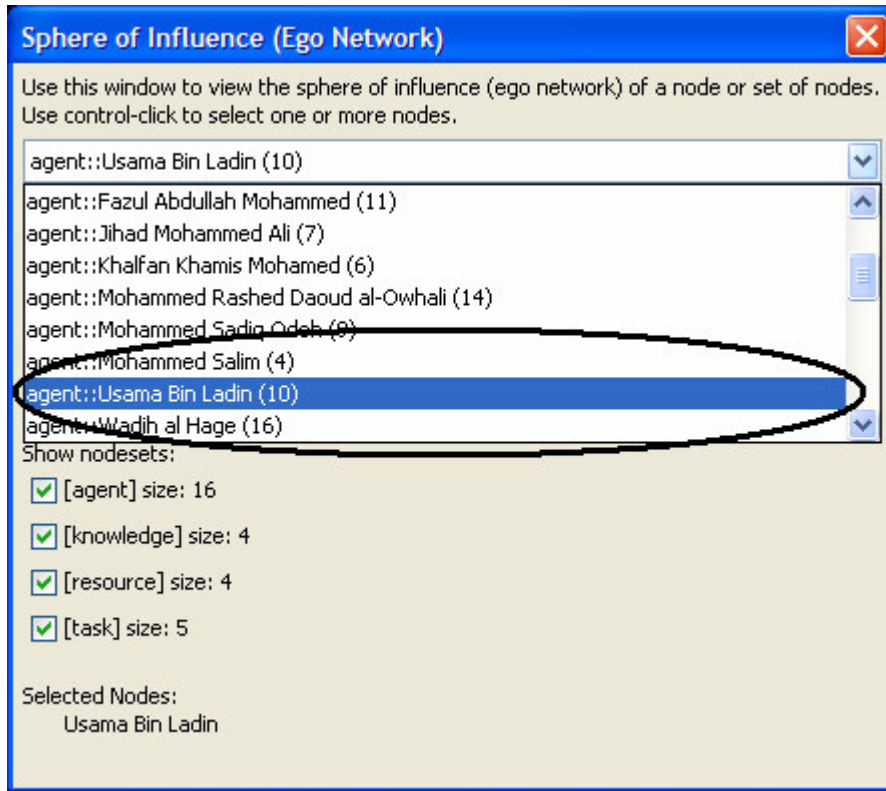
Sphere of Influence Example

Using the "Embassy" MetaMatrix as an example, we will find the Sphere of Influence of the Agent node "Usama Bin Ladin" by performing a basic analysis.

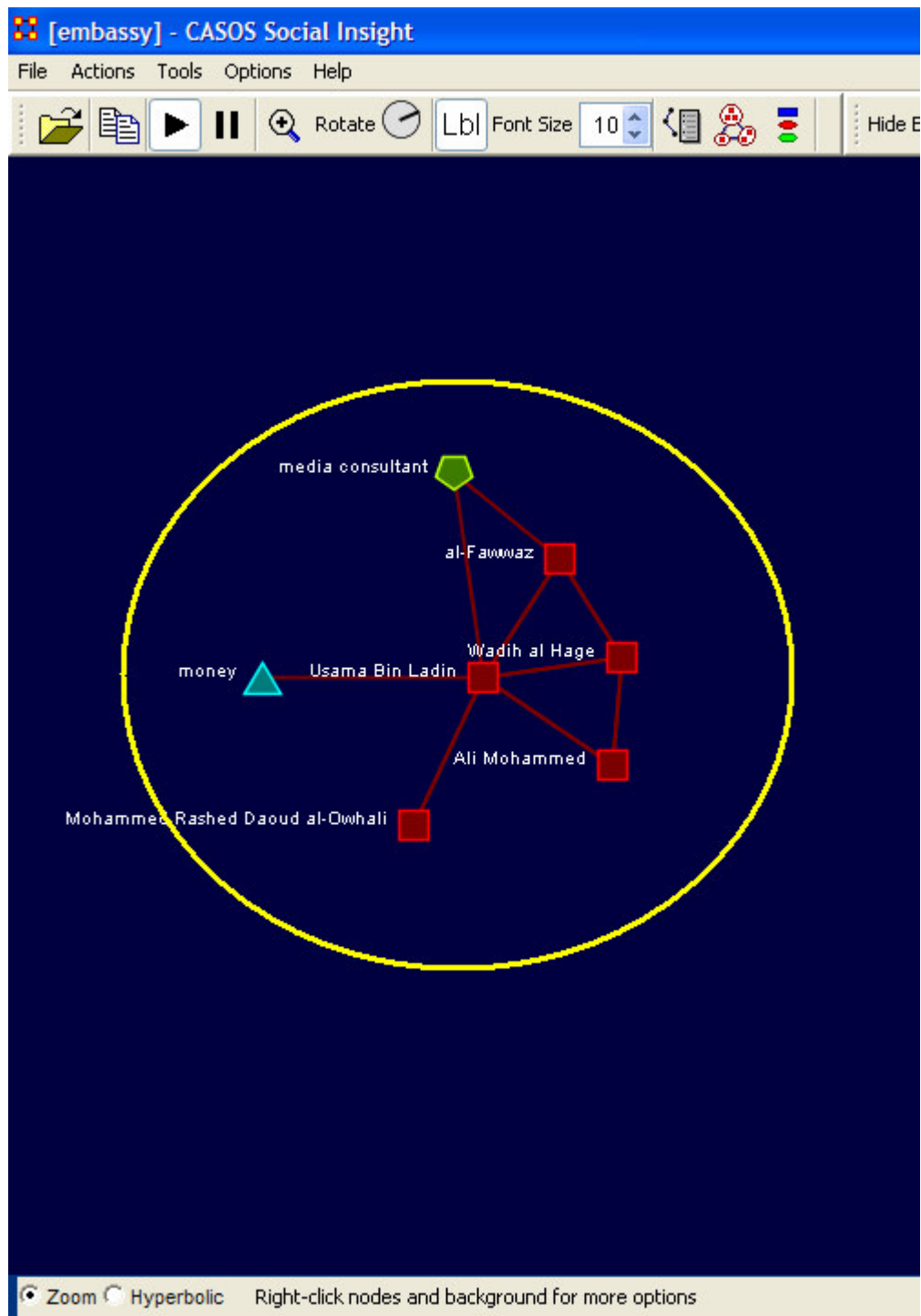
First, we load the "Embassy" MetaMatrix into the ORA Visualize.



Next, in the Sphere of Influence pop-up window (below) we select the node "Usama Bin Laden."

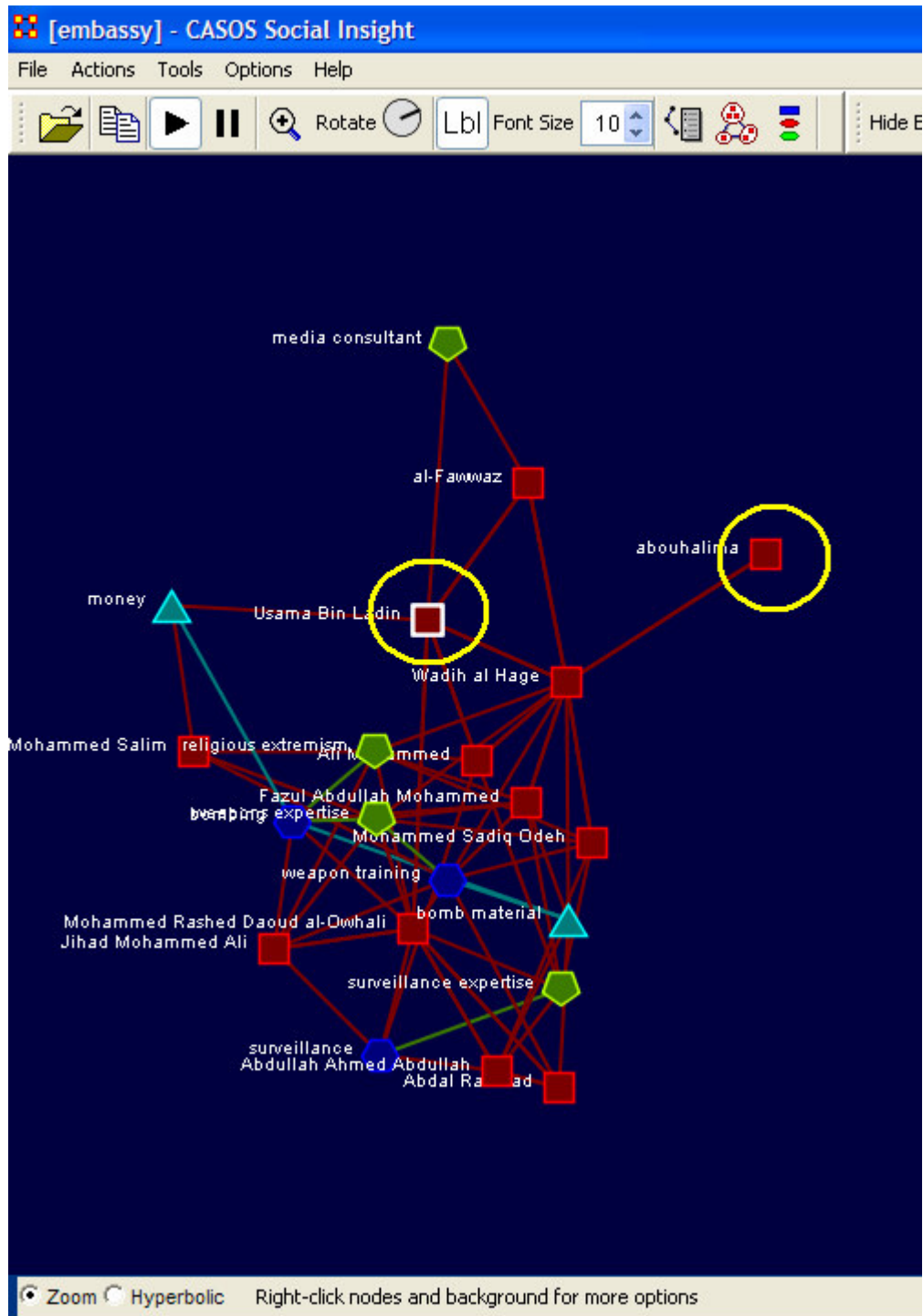


What follows is the resulting Ego Map of the agent "Usama Bin Ladin."



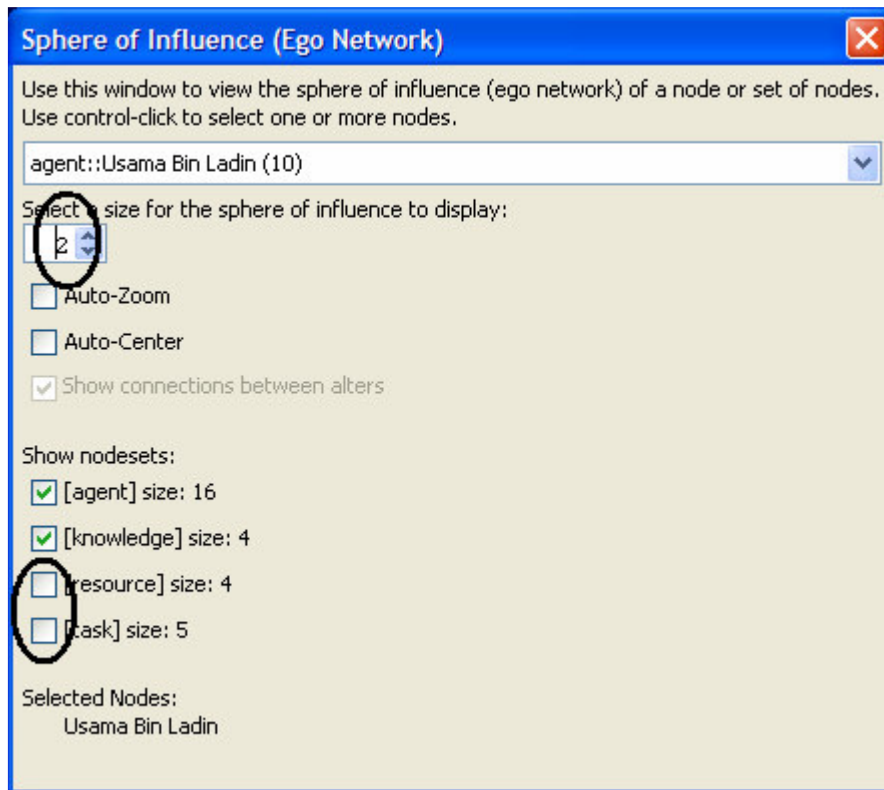
If you wanted, you can change the path length from "1" to "2" in the size-selector.

The yellow ellipses below highlight what happens when the path length is changed to "2." The nodes are not directly connected, but share an agent node, "Wadih al Hage," in common. Thus, we say they have a path length of "2."

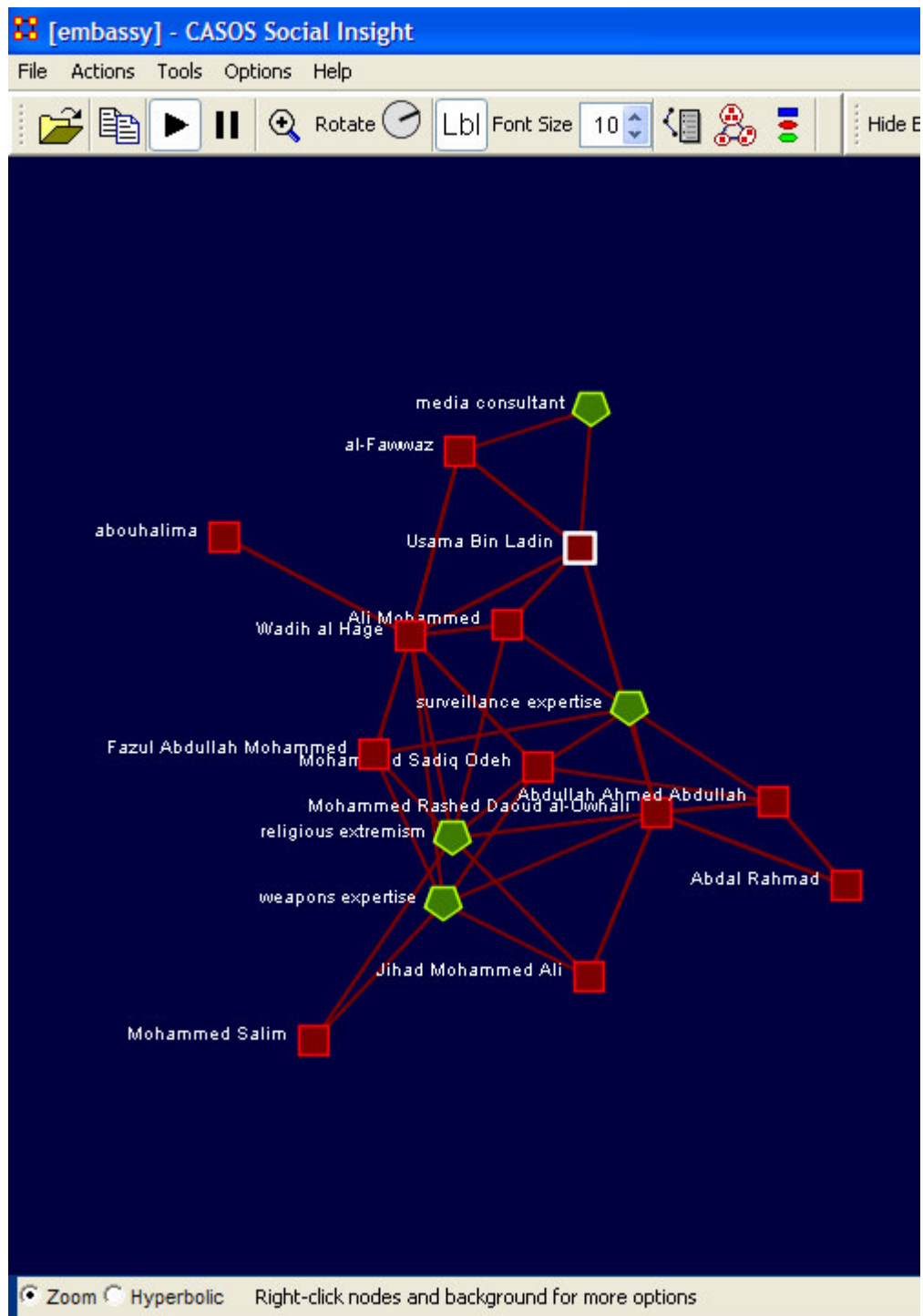


Let us examine the same Ego Network this time showing only the "knowledge-based" and "agent-based" nodes in relation to the agent node "Usama Bin Ladin."

To do this, in the Sphere of Influence window "de-select" the node sets we do not wish to include in our analysis.



The resulting Ego Map is shown below.

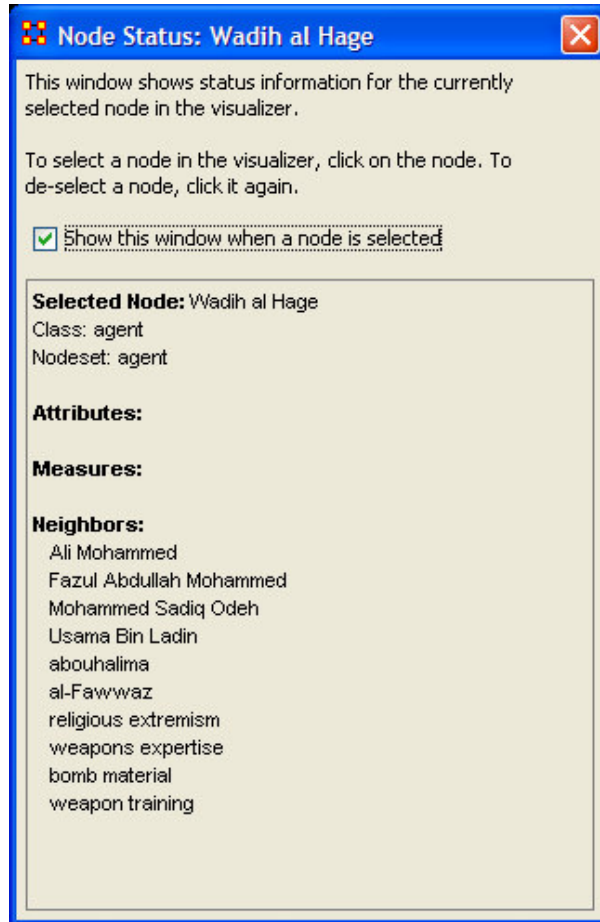


[Back to Sphere of Influence](#)

Node Status

The *Node Status Window* is selected by default. It first pops-up when you select any node in the Visualizer. This window provides you with a snapshot of the currently selected node by displaying that nodes unique *Attributes*, *Measures*, and *Neighbors*.

Below is a screen shot of the Node Status Window. By *un-checking* the box "Show this window when a node is selected," the Node Status Window will cease to appear when clicking on nodes within the Visualizer.



Attributes are descriptions you may or may not have given to particular node or nodeset in your network.

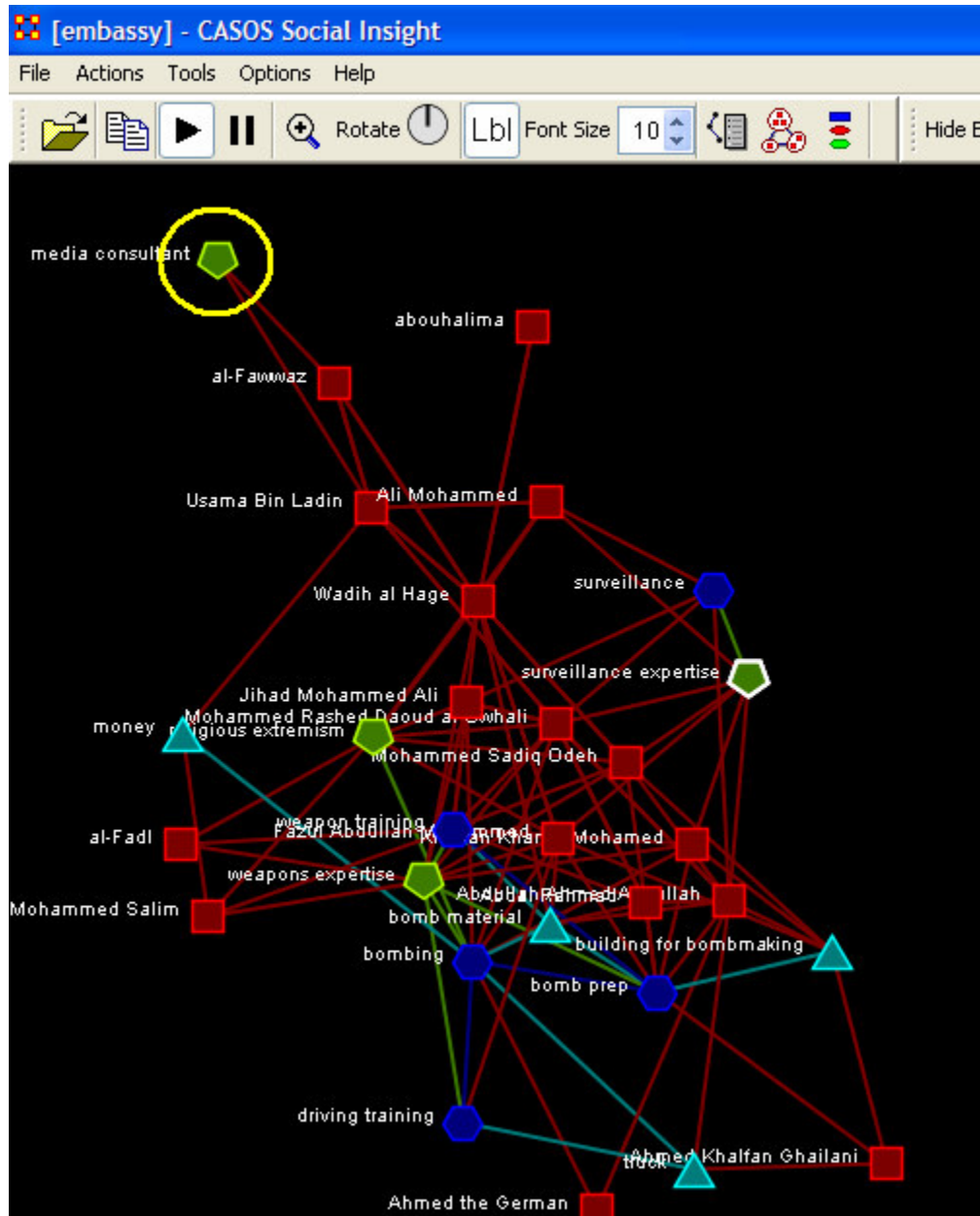
Measures describes computational functions in which the node is directly involved. See [Ora Measures](#) for more information about measures.

Neighbors are nodes that share an immediate link to the node selected.

[Node Status Example](#)

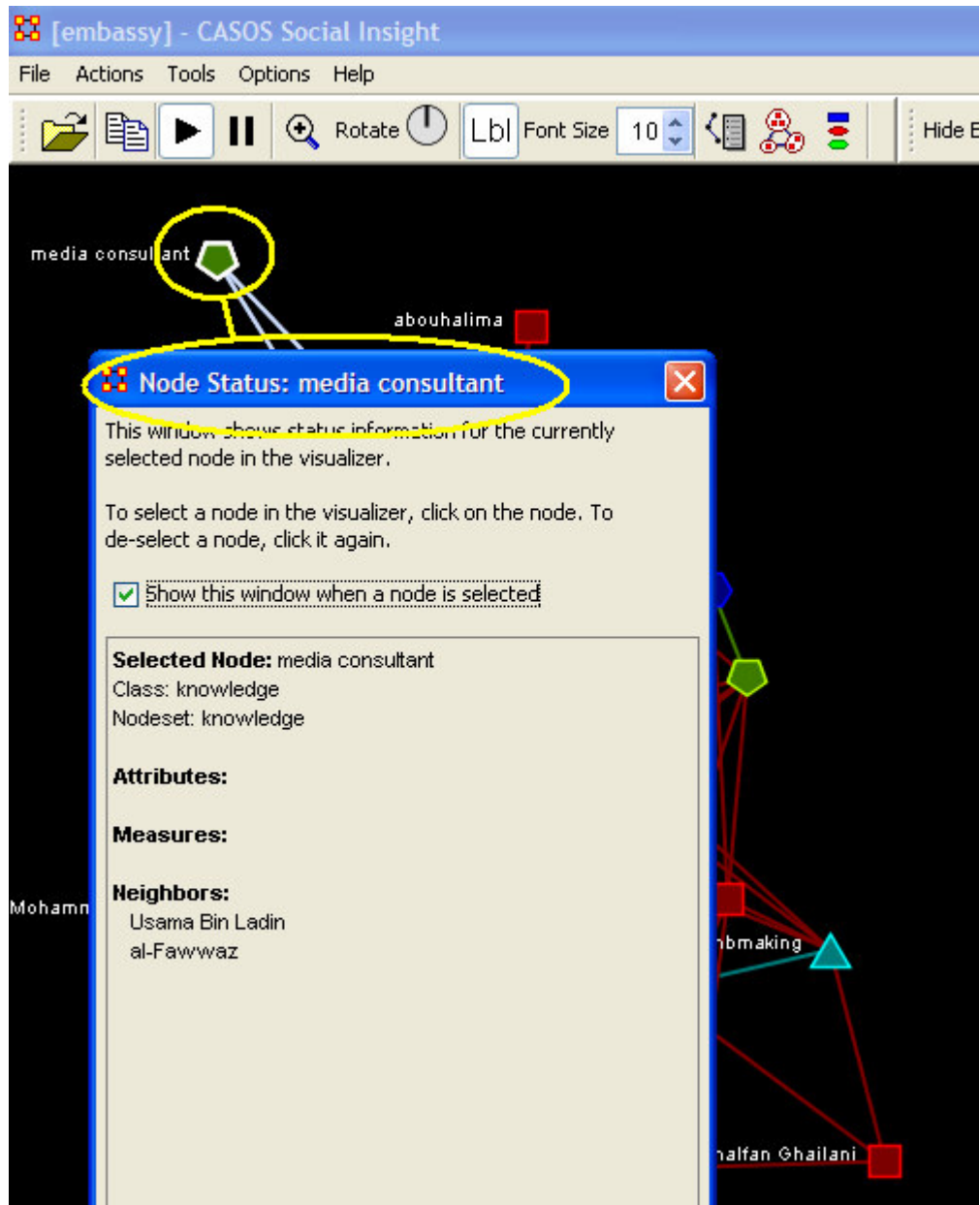
Node Status Example

Using the *Embassy MetaMatrix* visualization, simply click on the node *media consultant* highlighted in the following screenshot with the yellow ellipse.



The screen shot below displays the result. You will see that *media consultant* has two neighbors, *Usama Bin Laden* and *al-Fawwaz*.

Tip! If you are more interested in using other functions within the Visualizer, such as the Path Finder, deselect the Node Status option to un-clutter your visualization.



[Back to Node Status](#)

Actions

This folder contains help on using the "action" functions accessible from the Visualizer drop-down menu. Each help document will contain increasingly detailed information on using that function.

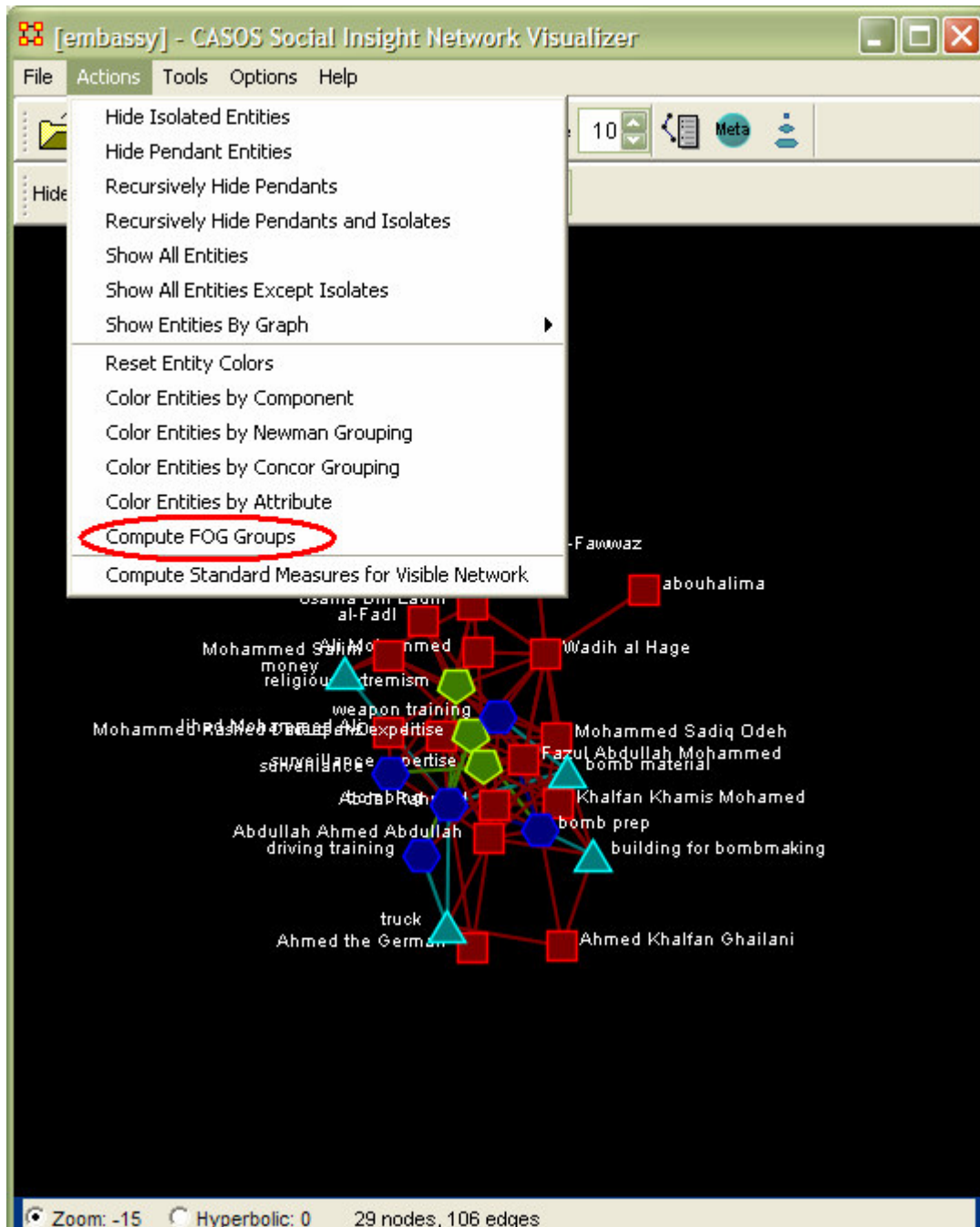
[Compute FOG Groups](#)

Compute FOG Groups

The ORA Visualizer can locate FOG Groups within your MetaMatrix. FOG (Fuzzy Overlapping Groups) indicates entities that can belong to groups with various strength and the likelihood those entities will participate in events associated with that group. Overlapping occurs when individuals belong to many groups simultaneously, so that the groups share members. To access the ORA Compute Fog Groups you must first be working in the Visualizer.

From the Visualizer drop-down tool bar: Actions > Compute FOG Groups

The red circle below highlights how to access the Compute FOG Groups function.

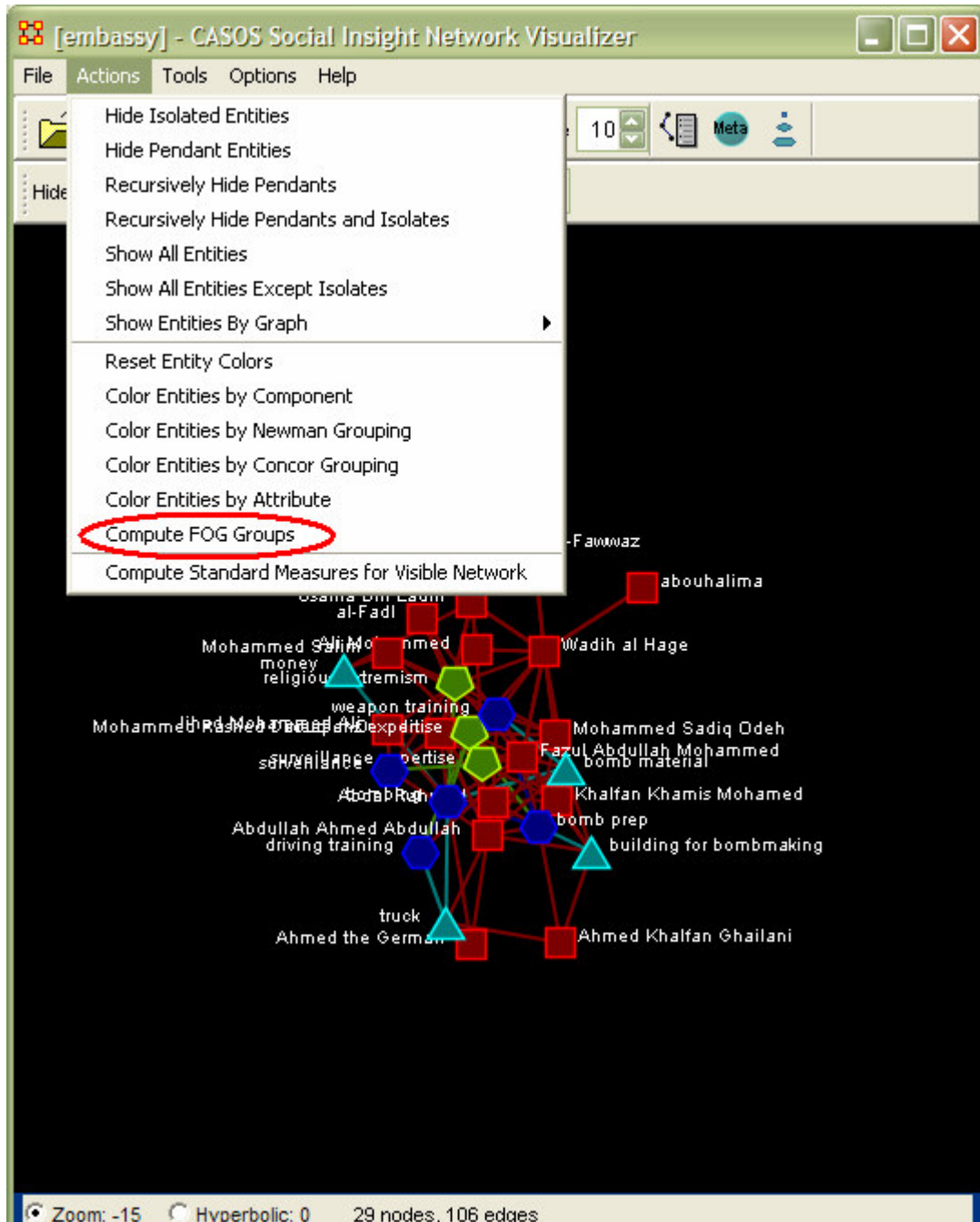


[Compute FOG Groups Example](#)

[Back to Actions](#)

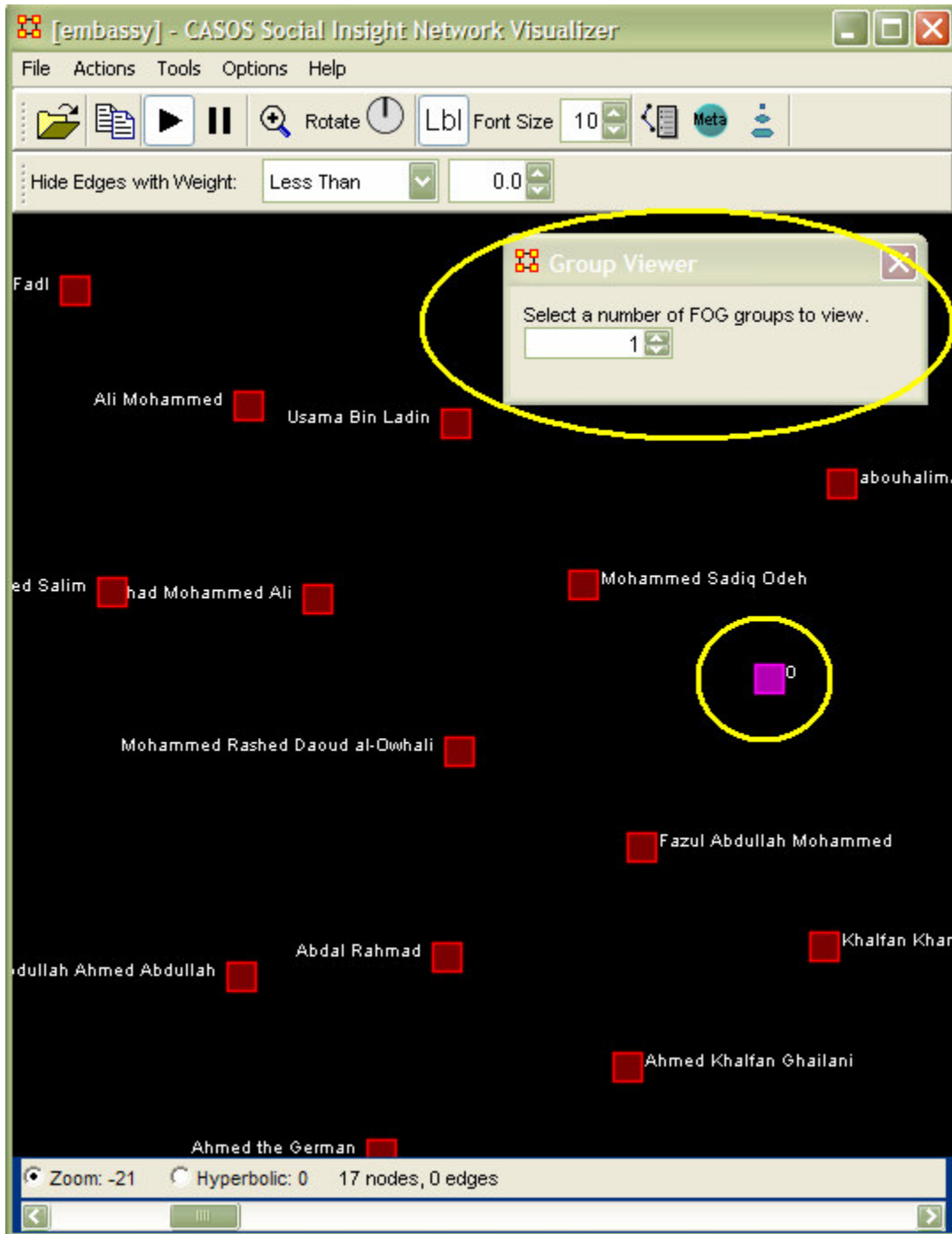
Compute FOG Group Example

Using the Embassy.xml MetaMatrix, we will use the Compute FOG Group function to determine FOG groups within our network. First, load in the Embassy.xml MetaMatrix, then access the Compute FOG Group function from the Visualizer drop-down tool bar. The red ellipse below highlights where to find the Compute FOG Group function.



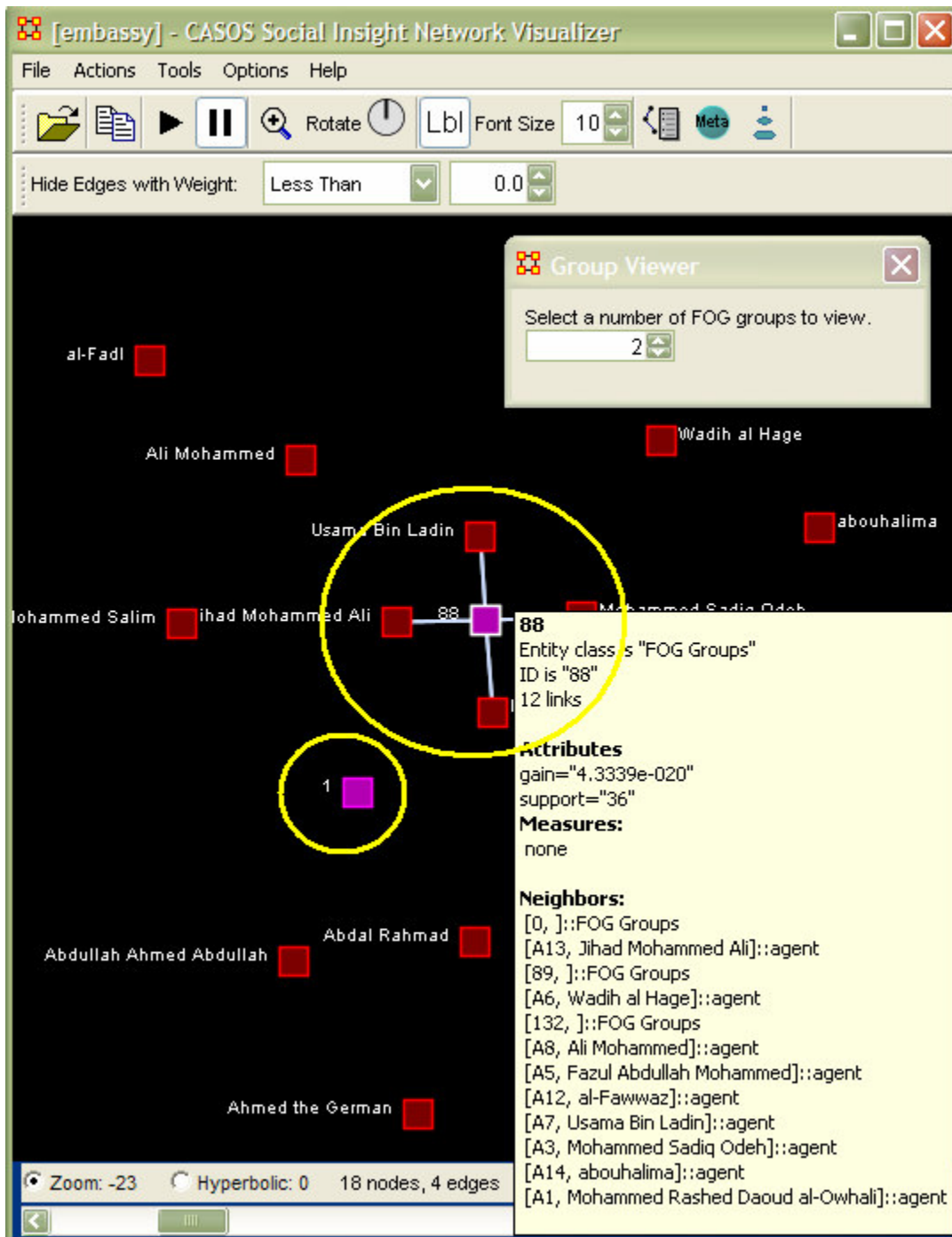
[Click Compute FOG Groups.](#)

The following Group Viewer selector box will appear. The yellow ellipse in the screen shot below shows the Group Viewer box. By default, the selector box will be set to "1." The first FOG Group will be displayed, denoted in the screen shot below as the purple square in the smaller yellow ellipse.



Select the number of FOG Groups to compute.

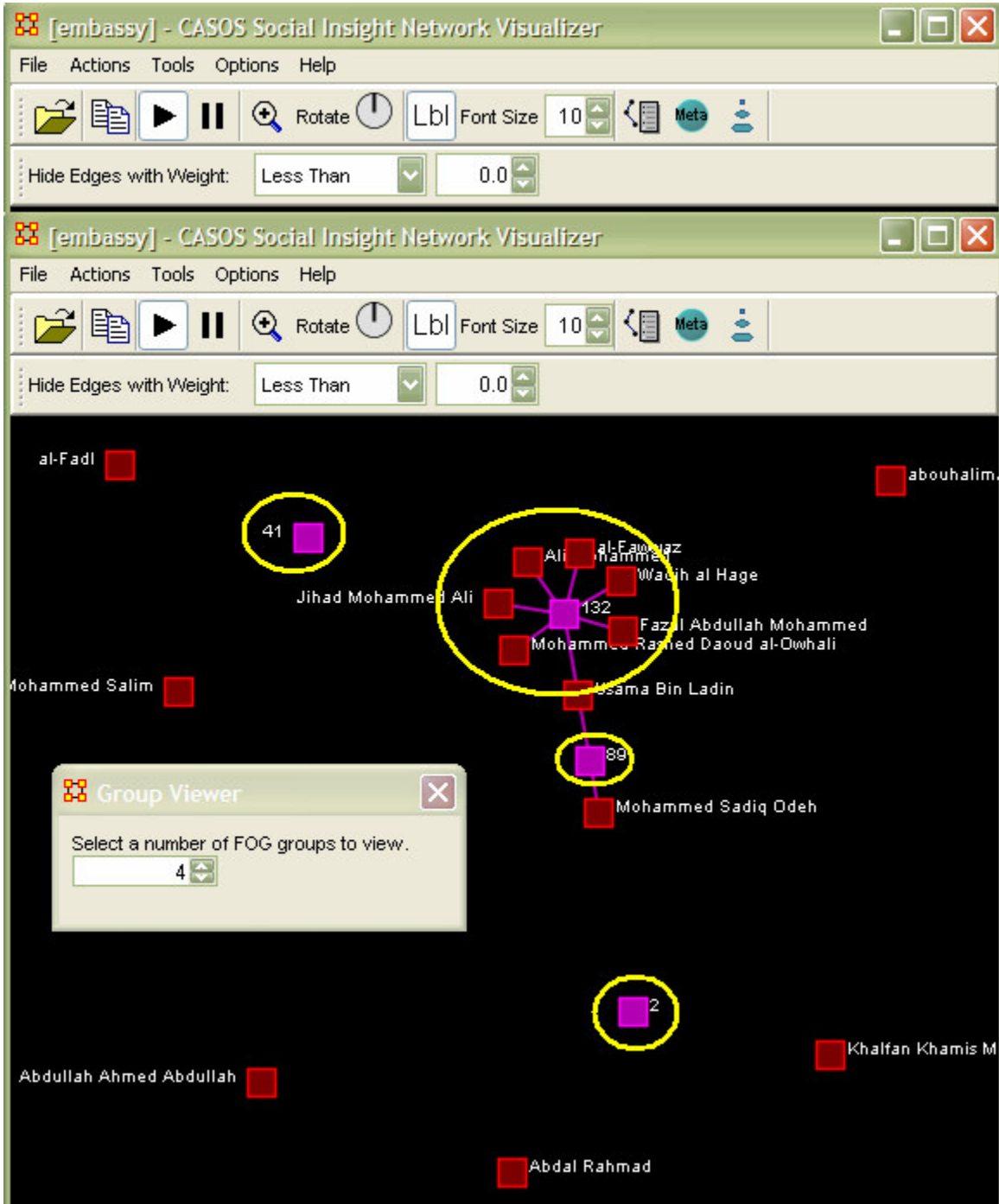
In this example, we will change the selector to "2." Notice the changes in the Visualizer. The screen shot below displays the Embassy.xml MetaMatrix with the Compute FOG Group selector changed to "2."



Two FOG Groups are displayed, denoted in the yellow ellipses in the screen shot above.

By clicking on any individual FOG Group you can access the Entity Status Box (shown above). This will list that FOG Groups Attributes, Measures, and Neighbors.

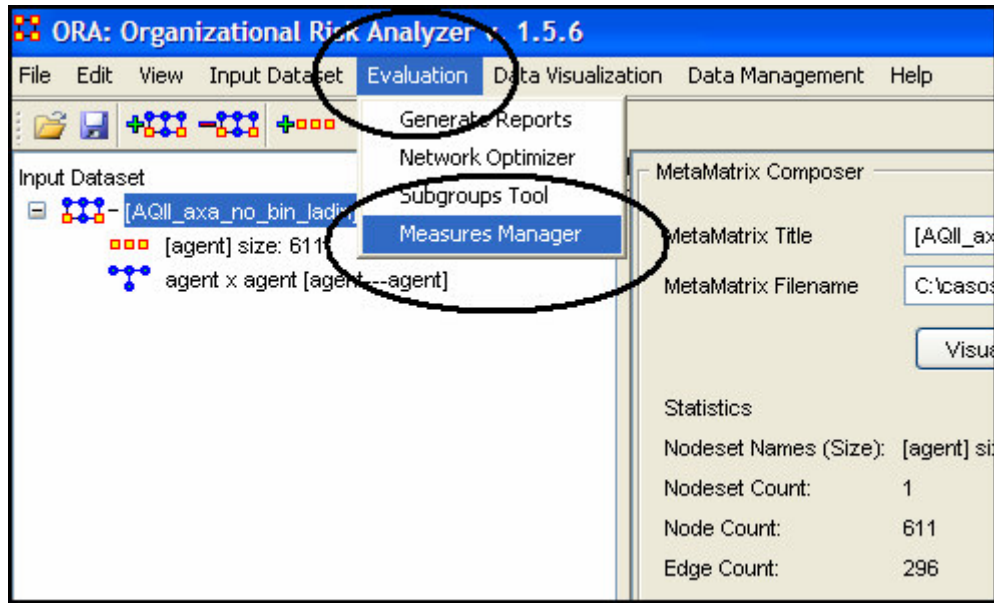
Finally, below is a screen shot with the Compute FOG Groups selector set to "3" using the Embassy.xml MetaMatrix as an example.



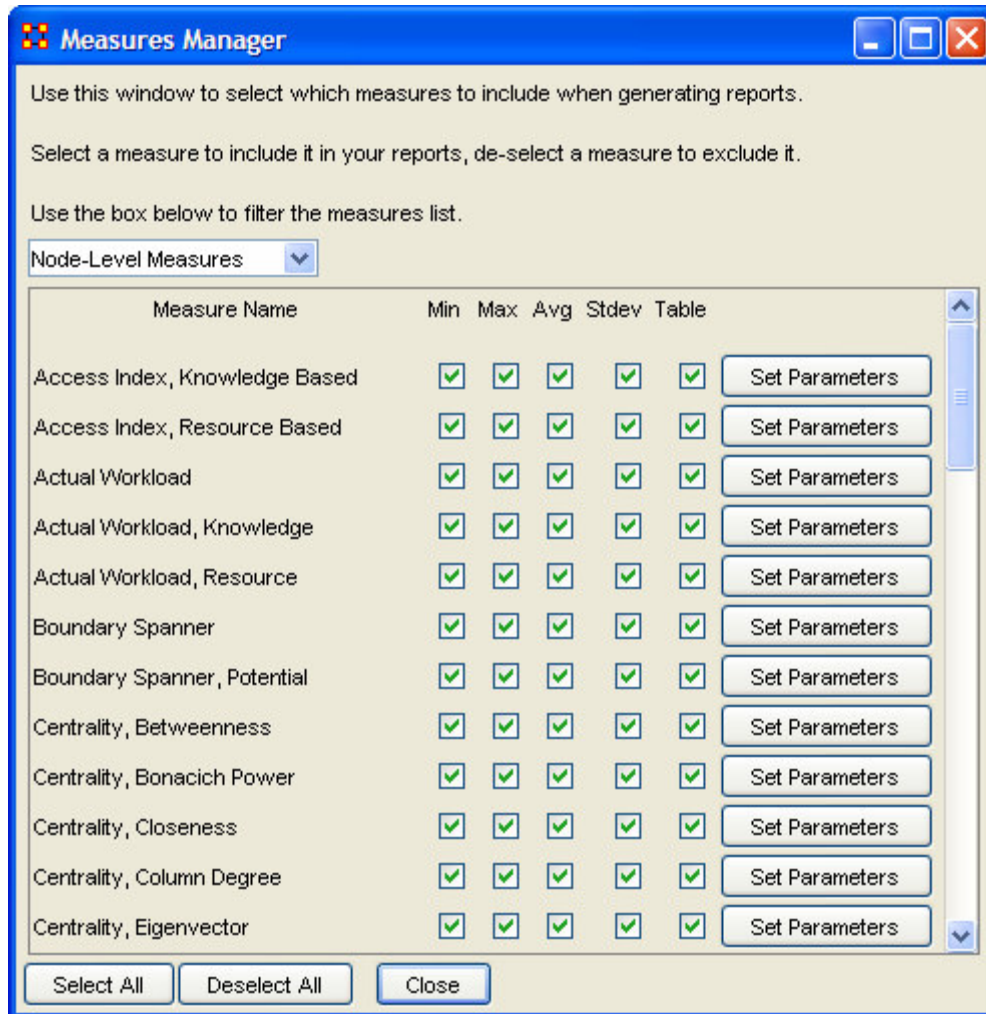
[Back To Actions](#)

ORA Measures

ORA contains over 100 measures. A measure is a function that takes as input a MetaMatrix and outputs a single value or a vector of values. Consider the measure "Density." The output for this measure is a single number used to analyze an organization. By default, all measures are run on a MetaMatrix. To view which measures are available, go to the Tools menu and select the Measures Manager. The black ellipses in the screen shot below highlight how to access the Measures Manager.



The Measures Manager dialog box will appear. A drop down box displays the measures in ORA.



The Measures Manager categorizes measures in the following ways: *Node Level*, *Graph Level*, and *Risk Category*.

A Node Level measure produces vector output, a single value per node. For example, "Betweenness Centrality" run on the Agent x Agent network outputs a value for each agent node. A Graph Level measure outputs a single value. For example, "Density" run on any network outputs a single number. Additionally, seven risk categories include measures describing certain aspects of your organization's structure. All of the measures can be selected and de-selected using the buttons at the bottom of the dialog box. When a measure is deselected, it will not be used when generating certain reports.

Some reports use a predefined set of measures, and these are not affected by the Measure Manager selections (for example, the *Intel*, *Context*, *Located SubGroups*, *Sphere of Influence*, and *Immediate Impact* reports). The Risk Report, however, uses only the measures selected in the Measure Manager.

The following sets of nodes (with their abbreviated symbol) are used throughout the document: Agent (**A**), Knowledge (**K**), Resource (**R**), and Task (**T**). The following networks defined on these node sets are used throughout the documentation:

Symbol	Node Sets		Name
	U	V	
AA	Agent	Agent	Communication Network
AK	Agent	Knowledge	Knowledge Network
AR	Agent	Resource	Capabilities Network
AT	Agent	Task	Assignment Network
KK	Knowledge	Knowledge	Information Network
KR	Knowledge	Resource	Training Network
KT	Knowledge	Task	Knowledge Requirement Network
RR	Resource	Resource	Resource Substitute Network
RT	Resource	Task	Resource Requirement Network
TT	Task	Task	Precedence Network

A complete list of all measures available in ORA, along with references, input and output specifications, can be found in the following ORA Measures sections.

The Bonacich Power Centrality

The Bonacich Power Centrality computes the centrality of each node based on the centrality of its neighbors. Beta should be chosen such that its absolute value is less than the reciprocal of the largest eigenvalue of N.

Bonachich, P, 1987

TYPE: Node Level

INPUT: N: Square, Beta

OUTPUT: Real Values between 0 and 1

Centrality, Eigenvector

Calculates the eigenvector of the largest positive eigenvalue of the adjacency matrix representation of a square network.

Measures the extent to which a node is connected to others who are also tightly connected to each other. Members of large cliques often have high eigenvector centrality. Note, this measure is only calculated for the nodes in the largest component. This is often thought of as a measure of power. That is, individuals high in eigenvector centrality are thought to have greater social capital to draw on and so more power.

Bonachich, P, 1972

TYPE: Node Level

INPUT: N: Square, Symmetric

OUTPUT: Real Values between 0 and 1

Centrality, Total Degree

The Total Degree Centrality of a node in a square network is its normalized In plus Out degree. The Total Degree Centrality of a node is the normalized sum of its row and column degrees.

Individuals who are high in total degree centrality tend to be “in the know” and have great access to information. Social extroverts often have high total degree centrality.

Wasserman and Faust, 1994 (pg 199)

TYPE: Node Level

INPUT: N: Square, Undirected

OUTPUT: Real Values between 0 and 1

Access Index, both Knowledge and Resource Based

Boolean value which is true if an agent is the only agent who knows a piece of knowledge and who is known by exactly one other agent. The one agent known also has its KAI set to one.

Ashworth, 2003

TYPE: Agent Level

INPUT: AK:binary; AA:binary

OUTPUT: Binary

Access Index, both Knowledge and Resource

The knowledge an agent uses to perform the tasks to which it is assigned.

Carley, 2002

TYPE: Agent Level

INPUT: AK:binary; KT:binary; AT:binary

OUTPUT: Binary

Boundary Spanner

A node which if removed from a network creates a new component. This is often called a Gate Keeper node.

Cormen, Leiserson, Rivest, Stein, 2001 p.558

TYPE: Agent Level

INPUT: N: square, symmetric

OUTPUT: Binary

Centrality, Betweenness

The Betweenness Centrality of node v in a network is defined as: across all node pairs that have a shortest path containing v , the percentage that pass through v . This is defined for directed networks.

Freeman, 1979

TYPE: Node Level

INPUT: N: square

OUTPUT: Real Values between 1 and 0

Centrality, Bonacich Power

The Bonacich Power Centrality computes the centrality of each node based on the centrality of its neighbors. β should be chosen such that its absolute value is less than the reciprocal of the largest eigenvalue of N .

Bonacich P, 1987

TYPE: Node Level

INPUT: N: square, β is a real value

OUTPUT: Real Values between 1 and 0

Centrality, Closeness

The average closeness of a node to the other nodes in a network. Loosely, Closeness is the inverse of the average distance in the network between the node and all other nodes. This is defined for directed networks.

Freeman, 1979

TYPE: Node Level

INPUT: N: square

OUTPUT: Real Values between 1 and 0

Centrality, Eigenvector

Calculates the eigenvector of the largest positive eigenvalue of the adjacency matrix representation of a square network.

Bonacich P, 1972

TYPE: Node Level

INPUT: N: square, symmetric

OUTPUT: Real Values between 1 and 0

Centrality, In Degree

The In Degree Centrality of a node in a unimodal network is its normalized In-degree.

Wasserman and Faust, 1994

TYPE: Node Level

INPUT: N: square

OUTPUT: Real Values between 1 and 0

Centrality, Information

Calculate the Stephenson and Zelen information centrality measure for each node.

Wasserman and Faust, 1994 (pg. 195)

TYPE: Node Level

INPUT: N:square, symmetric

OUTPUT: Real Values between 1 and 0

Centrality, Inverse Closeness

The average closeness of a node to the other nodes in a network. Inverse Closeness is the sum of the inverse distances between a node and all other nodes. This is defined for directed networks.

Wasserman and Faust, 1994 (pg 195)

TYPE: Node Level

INPUT: N:square

OUTPUT: Real Values between 1 and 0

Centrality, Out Degree

The Out Degree Centrality of a node in a square network is its normalized out-degree.

Wasserman and Faust, 1994

TYPE: Node Level

INPUT: N:square

OUTPUT: Real Values between 1 and 0

Centrality, Total Degree

The Total Degree Centrality of a node in a square network is its normalized in plus out degree. The Total Degree Centrality of a node is the normalized sum of its row and column degrees.

Individuals who are high in total degree centrality tend to be “in the know” and have great access to information. Social extroverts often have high total degree centrality.

Wasserman and Faust, 1994 (pg 254)

TYPE: Node Level

INPUT: N: square, undirected

OUTPUT: Integer, unscaled

Clique Count

Computes the number of distinct cliques to which each node in a square, undirected network belongs.

Wasserman and Faust, 1994 (pg 254)

TYPE: Node Level

INPUT: N: square, undirected

OUTPUT: Real value between 0 and 1

Clustering Coefficient, Watts-Strogatz

Measures the degree of clustering in a network by averaging the clustering coefficient of each node i . The clustering coefficient of a node is the density of its ego network (which is the sub graph induced by its immediate neighbors).

Watts and Strogatz, 1998

TYPE: Graph Level, Node Level

INPUT: N: square

OUTPUT: Real value between 0 and 1

Cognitive Demand

Measures the total amount of effort expended by each agent to do its tasks. Measures the total cognitive effort expended by an agent to do its tasks.

Individuals who are high in cognitive demand are emergent leaders. Removal of these individuals is quite disruptive to networks.

Note: The minimum input requirement is the AA network. All other networks are optional.

Carley, 2002

TYPE: Agent Level

INPUT: AA:binary; [AT:binary]; [AR:binary]; [RT:binary]; [AK:binary]; [KT:binary]; [TT:binary]

OUTPUT: Real value between 0 and 1

Cognitive Distinctiveness

Measures how distinct are two agents based on the number of knowledge bits they hold oppositely.

Carley, 2002

TYPE: Dyad Level

INPUT: AK: binary

OUTPUT: Real value between 0 and 1

Relative Cognitive Distinctiveness

Measures how distinct are two agents based on the number of knowledge bits they hold oppositely.

Carley, 2002

TYPE: Dyad Level

INPUT: AK: binary

OUTPUT: Real value between 0 and 1

Cognitive Expertise

Measures the complementarity of two agents based on their knowledge.

Carley, 2002

TYPE: Dyad Level

INPUT: AK: binary

OUTPUT: Real value between 0 and 1

Relative Cognitive Expertise

Measures the complementarity of two agents based on their knowledge.

Carley, 2002

TYPE: Dyad Level

INPUT: AK: binary

OUTPUT: Real value between 0 and 1

Cognitive Resemblance

Measures the degree of resemblance between agents based on the number of knowledge bits they both have or both do not have.

Carley, 2002

TYPE: Dyad Level

INPUT: AK: binary

OUTPUT: Real value between 0 and 1

Cognitive Similarity

Measures the degree of similarity between agents based on the number of knowledge bits they both have.

Carley, 2002

TYPE: Dyad Level

INPUT: AK: binary

OUTPUT: Real value between 0 and 1

Relative Cognitive Similarity

Measures the degree of similarity between agents based on the number of knowledge bits they both have.

Carley, 2002

TYPE: Dyad Level

INPUT: AK: binary

OUTPUT: Real value between 0 and 1

Communication

Measures the degree of similarity between agents based on the number of knowledge bits they both have.

Carley, 2003

TYPE: Agent Level

INPUT: AA:binary; AT:binary; AR:binary; RT:binary, TT:binary

OUTPUT: Real value between 0 and 1

Component Count, Strong

The number of strongly connected components in a network.

Wasserman and Faust, 1994 (pg 109)

TYPE: Graph Level

INPUT: N:square

OUTPUT: Integer value between 0 and $|v|$

Component Count, Weak

The number of weakly connected components in a network.

Wasserman and Faust, 1994 (pg 109)

TYPE: Graph Level

INPUT: N:square, symmetric

OUTPUT: Integer value between 0 and $|v|$

Component Members, Weak

Assigns each node an integer which corresponds to the weak component in the network to which it belongs.

Wasserman and Faust, 1994

TYPE: Node Level

INPUT: N:square, symmetric

OUTPUT: Integer value between 0 and $|v|$

Congruence, Agent Knowledge Needs

The number of skills that an agent lacks to complete its assigned tasks expressed as a percentage of the total skills required for the assigned tasks.

Lee, 2004

TYPE: Agent Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Agent Resource Needs

The number of skills that an agent lacks to complete its assigned tasks expressed as a percentage of the total skills required for the assigned tasks.

Lee, 2004

TYPE: Agent Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Agent Knowledge Waste

The number of skills that an agent has that are not needed by any of its tasks expressed as a percentage of the total skills of the agent.

Lee, 2004

TYPE: Agent Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Agent Resource Needs

The number of skills that an agent has that are not needed by any of its tasks expressed as a percentage of the total skills of the agent.

Lee, 2004

TYPE: Agent Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Communication

Measures to what extent the agents communicate when and only when it is useful to complete tasks. Perfect congruence requires a symmetric agent communication.

Carley, 2002

TYPE: Graph Level

INPUT: AA:binary; AT:binary; AR:binary; RT:binary, TT:binary

OUTPUT: Real value between 0 and 1

Congruence, Organization Agent Knowledge Needs

Across all agents, the skills that agents lack to do their assigned tasks expressed as a percentage of the total skills needed by all agents.

Lee, 2004

TYPE: Graph Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Organization Agent Resource Needs

Across all agents, the skills that agents lack to do their assigned tasks expressed as a percentage of the total skills needed by all agents.

Lee, 2004

TYPE: Graph Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Organization Agent Knowledge Waste

Across all agents, the skills that agents have that are not required to do their assigned tasks.

Lee, 2004

TYPE: Graph Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Organization Agent Resource Waste

Across all agents, the skills that agents have that are not required to do their assigned tasks.

Lee, 2004

TYPE: Graph Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Organization Task Knowledge Needs

Across all tasks, the skills that tasks lack expressed as a percentage of the total skills needed by all tasks.

Lee, 2004

TYPE: Graph Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Organization Task Resource Needs

Across all tasks, the skills that tasks lack expressed as a percentage of the total skills needed by all tasks.

Lee, 2004

TYPE: Graph Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Organization Task Knowledge Waste

Across all tasks, the skills supplied to tasks via agents that are not required by them, expressed as a percentage of the total skills needed by all tasks.

Lee, 2004

TYPE: Graph Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Organization Task Resource Waste

Across all tasks, the skills supplied to tasks via agents that are not required by them, expressed as a percentage of the total skills needed by all tasks.

Lee, 2004

TYPE: Graph Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Strict Knowledge

Measures the similarity between what knowledge is assigned to tasks via agents, and what knowledge is required to do tasks. Perfect congruence occurs when agents have knowledge when and only when (strictly) it is needful to complete tasks.

Carley, 2002

TYPE: Graph Level

INPUT: AK:binary; AT:binary; KT:binary

OUTPUT: Real value between 0 and 1

Congruence, Strict Resource

Measures the similarity between what knowledge is assigned to tasks via agents, and what knowledge is required to do tasks. Perfect congruence occurs when agents have knowledge when and only when (strictly) it is needful to complete tasks.

Carley, 2002

TYPE: Graph Level

INPUT: AK:binary; AT:binary; KT:binary

OUTPUT: Real value between 0 and 1

Congruence, Task Knowledge Needs

The number of skills not supplied to a task, and required to do the task, expressed as a percentage of the total skills required for the task.

Carley, 2002

TYPE: Task Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Task Resource Needs

The number of skills not supplied to a task, and required to do the task, expressed as a percentage of the total skills required for the task.

Carley, 2002

TYPE: Task Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Task Knowledge Waste

The number of skills supplied to a task via agents that are not required by it expressed as a percentage of the total skills required for the task.

Carley, 2002

TYPE: Task Level

INPUT: AK/AR:binary; KT/RT:binary; AT:binary

OUTPUT: Real value between 0 and 1

Congruence, Task Resource Waste

The number of skills supplied to a task via agents that are not required by it expressed as a percentage of the total skills required for the task.

Carley, 2002

TYPE: Task Level

INPUT: AK/AR: binary; KT/RT: binary; AT: binary

OUTPUT: Real value between 0 and 1

Connectedness, Krackhardt

Measures the degree to which a square network's underlying (undirected) network is connected.

Krackhardt, 1994

TYPE: Graph Level

INPUT: N:square, symmetric

OUTPUT: Real value between 0 and 1

Constraint, Burt

The degree to which each node in a square network is constrained from acting because of its existing links to other nodes.

Burt, 1992

TYPE: Node Level

INPUT: N:square

OUTPUT: Real value between 0 and 1

Density

The ratio of the number of edges versus the maximum possible edges for a network.

Wasserman and Faust, 1994 (pg 101)

TYPE: Graph Level

INPUT: N

OUTPUT: Real value between 0 and 1

Diameter

The maximum shortest path length between any two nodes in a unimodal network $G=(V,E)$. If there exist i,j in V such that j is not reachable from i , then $|V|$ is returned.

Wasserman and Faust, 1994 (pg 111)

TYPE: Graph Level

INPUT: N:square

OUTPUT: Integer value between 0 and $|V|$

Distance Weighted Reach

A generalization of graph theoretic distance, this measures the distance from a set of nodes in the network to all other nodes.

Borgatti, 2003

TYPE: Graph Level

INPUT: N:square, undirected

OUTPUT: Real value between 0 and 1

Diversity, Knowledge

The distribution of difference in idea sharing. This is the Herfindahl-Hirshman index applied to column sums of AK.

Borgatti, 2003

TYPE: Graph Level

INPUT: AK:binary

OUTPUT: Real value between 0 and 1

Diversity, Resource

The distribution of difference in idea sharing. This is the Herfindahl-Hirshman index applied to column sums of AK.

Borgatti, 2003

TYPE: Graph Level

INPUT: AK:binary

OUTPUT: Real value between 0 and 1

Edge Count, Lateral

The percentage of lateral edges in a network. Fixing a root node x , a lateral edge (i,j) is one in which the distance from x to i is the same as the distance from x to j .

Carley, 2002

TYPE: Graph Level

INPUT: N:square

OUTPUT: Real value between 0 and 1

Edge Count, Lateral

The percentage of lateral edges in a network. Fixing a root node x , a lateral edge (i,j) is one in which the distance from x to i is the same as the distance from x to j .

Carley, 2002

TYPE: Graph Level

INPUT: N:square

OUTPUT: Real value between 0 and 1

Edge Count, Reciprocal

The percentage of edges in a network that are reciprocated (also called Reciprocity). An edge (i,j) in the network is reciprocated if edge (j,i) is also in the network. Self-loops are ignored.

Carley, 2002

TYPE: Graph Level

INPUT: N: square

OUTPUT: Real value between 0 and 1

Edge Count, Sequential

The percentage of edges in a unimodal network that are neither Reciprocal Edges nor Pooled Edges. Note that an edge can be both a Pooled and a Reciprocal edge. Self-loops are ignored.

Carley, 2002

TYPE: Graph Level

INPUT: N: square

OUTPUT: Real value between 0 and 1

Edge Count, Skip

The fraction of edges in a unimodal network that skip levels. An edge (i,j) is a skip edge if there is a path from node i to node j even after the edge (i,j) is removed.

Carley, 2002

TYPE: Graph Level

INPUT: N: square

OUTPUT: Real value between 0 and 1

Effective Network Size

The effective size of a node's ego network based on redundancy of ties.

Burt, 1992

TYPE: Node Level

INPUT: N: square

OUTPUT: Real value between 0 and 1

Efficiency, Global

Measures the closeness of the nodes in the network.

Latora and Marchiori, 2001

TYPE: Graph Level

INPUT: N: square, symmetric

OUTPUT: Real value between 0 and 1. This routine symmetrizes the input graph.

Efficiency, Krackhardt

The degree to which each component in a network contains the minimum edges possible to keep it connected.

Krackhardt, 1994

TYPE: Graph Level

INPUT: N: square, symmetric

OUTPUT: Real value between 0 and 1. This routine symmetrizes the input graph.

Efficiency, Local

Measures the closeness of the nodes in each ego network in the network.

Latora and Marchiori, 2001

TYPE: Graph Level

INPUT: N: square, symmetric

OUTPUT: Real value between 0 and 1. This routine symmetrizes the input graph.

Exclusivity, Knowledge

Detects agents who have singular knowledge.

Ashworth, 2003

TYPE: Agent Level

INPUT: AK: binary

OUTPUT: Real value between 0 and 1.

Exclusivity, Resource

Detects agents who have singular knowledge.

Ashworth, 2003

TYPE: Agent Level

INPUT: AK: binary

OUTPUT: Real value between 0 and 1.

Exclusivity, Task

Detects agents who have singular knowledge.

Ashworth, 2003

TYPE: Agent Level

INPUT: AK:binary

OUTPUT: Real value between 0 and 1.

Fragmentation

The proportion of nodes in a network that are disconnected.

Borgatti, 2003

TYPE: Graph Level

INPUT: N:square, undirected

OUTPUT: Real value between 0 and 1.

Hierarchy, Krackhardt

The degree to which a unimodal network exhibits a pure hierarchical structure.

Krackhardt, 1994

TYPE: Graph Level

INPUT: N:square

OUTPUT: Real value between 0 and 1.

Interdependence

The percentage of edges in a unimodal network that are Pooled or Reciprocal.

Carley, 2002

TYPE: Graph Level

INPUT: N: square

OUTPUT: Real value between 0 and 1.

Interlockers

Interlocker and radial nodes in a square network have a high and low Triad Count, respectively.

Carley, 2002

TYPE: Node Level

INPUT: N: square

OUTPUT: Binary

Radials

Interlocker and radial nodes in a square network have a high and low Triad Count, respectively.

Carley, 2002

TYPE: Node Level

INPUT: N: square

OUTPUT: Binary

Load, Knowledge

Average number of knowledge per agent.

Carley, 2002

TYPE: Graph Level

INPUT: AK:binary

OUTPUT: Real value between 0 and | K |

Load, Resource

Average number of knowledge per agent.

Carley, 2002

TYPE: Graph Level

INPUT: AK:binary

OUTPUT: Real value between 0 and | K |

Negotiation, Knowledge

The extent to which agents need to negotiate with each other because they lack the knowledge to complete their assigned tasks.

Carley, 2002

TYPE: Graph Level

INPUT: AT:binary; AK:binary; KT:binary

OUTPUT: Real value between 0 and 1

Negotiation, Resource

The extent to which agents need to negotiate with each other because they lack the knowledge to complete their assigned tasks.

Carley, 2002

TYPE: Graph Level

INPUT: AT:binary; AK:binary; KT:binary

OUTPUT: Real value between 0 and 1

Network Centralization, Betweenness

Network centralization based on the betweenness score for each node in a square network. This measure is defined for directed and undirected networks.

Freeman, 1979

TYPE: Graph Level

INPUT: N:square

OUTPUT: Real value between 0 and 1

Network Centralization, Closeness

Network centralization based on the closeness centrality of each node in a square network. This is defined only for connected, undirected networks.

Freeman, 1979

TYPE: Graph Level

INPUT: N:square, symmetric, connected

OUTPUT: Real value between 0 and 1

Network Centralization, Column Degree

A centralization based on the degree of the column nodes of a network.

NetStat

TYPE: Graph Level

INPUT: N

OUTPUT: Real value between 0 and 1

Network Centralization, In Degree

A centralization of a square network based on the In-Degree Centrality of each node.

NetStat

TYPE: Graph Level

INPUT: N: square

OUTPUT: Real value between 0 and 1

Network Centralization, Out Degree

A centralization of a square network based on the Out-Degree Centrality of each node.

NetStat

TYPE: Graph Level

INPUT: N: square

OUTPUT: Real value between 0 and 1

Network Centralization, Row Degree

A centralization based on the degree of the row nodes in a network.

NetStat

TYPE: Graph Level

INPUT: N

OUTPUT: Real value between 0 and 1

Network Centralization, Total Degree

A centralization of a square network based on total degree centrality of each node.

Freeman, 1979

TYPE: Graph Level

INPUT: N:square, undirected

OUTPUT: Real value between 0 and 1

Network Levels

The Network Level of a square network is the maximum Node Level of its nodes.

NetStat

TYPE: Graph Level

INPUT: N:square

OUTPUT: Integer between 0 and $|V| - 1$

Node Levels

The Node Level for a node v in a square network is the longest shortest path from v to every node v can reach. If v cannot reach any node, then its level is 0.

Carley, 2002

TYPE: Node Level

INPUT: N:square

OUTPUT: Integer between 0 and $|V| - 1$

Omega, Knowledge

The degree to which agents reuse knowledge while doing their tasks

Carley, Dekker, and Krackhardt 2000

TYPE: Graph Level

INPUT: AT:binary; KT:binary; TT:binary

OUTPUT: Real value between 0 and 1

Omega, Resource

The degree to which agents reuse a resource while doing their tasks.

Carley, Dekker, and Krackhardt 2000

TYPE: Graph Level

INPUT: AT:binary; KT:binary; TT:binary

OUTPUT: Real value between 0 and 1

Performance as Accuracy

Measures how accurately agents can perform their assigned tasks based on their access to knowledge and resources.

Carley, 2002

TYPE: Graph Level

INPUT: AT:binary; AK:binary; AR:binary; KT:binary; RT:binary

OUTPUT: Real value between 0 and 1

Personnel Cost

Total number of people reporting to an agent, plus its total knowledge, resources, and tasks.

Carley, 2003

TYPE: Agent Level

INPUT: AA:binary; AK:binary; AR:binary; AT:binary

OUTPUT: Real value between 0 and 1

Potential Workload, Knowledge

Maximum knowledge an agent could use to do tasks if it were assigned to all tasks.

Carley, 2002

TYPE: Agent Level

INPUT: AK:binary; KT:binary

OUTPUT: Real value between 0 and 1

Potential Workload, Resource

Maximum knowledge an agent could use to do tasks if it were assigned to all tasks.

Carley, 2002

TYPE: Agent Level

INPUT: AK:binary; KT:binary

OUTPUT: Real value between 0 and 1

Redundancy, Access

Average number of redundant agents per resource. An agent is redundant if there is already an agent that has access to the resource.

Carley, 2002

TYPE: Graph Level

INPUT: AR:binary

OUTPUT: Real value between 0 and $(|A| - 1) \times R$

Redundancy, Assignment

Average number of redundant agents assigned to tasks. An agent is redundant if there is already an agent assigned to the task.

Carley, 2002

TYPE: Graph Level

INPUT: AT

OUTPUT: Real value between 0 and $(|A| - 1) \times T$

Redundancy, Column

The mean number of column node edges in excess of one.

Netstat

TYPE: Graph Level

INPUT: N of dimension $m \times n$

OUTPUT: Real value between 0, $(M-1) \times M$

Redundancy, Knowledge

Average number of redundant agents per knowledge. An agent is redundant if there is already an agent that has the knowledge.

Carley, 2002

TYPE: Graph Level

INPUT: AK

OUTPUT: Real value between 0, $(A-1) \times |K|$

Redundancy, Resource

Average number of redundant resources assigned to tasks. A resource is redundant if there is already a resource assigned to the task.

Carley, 2002

TYPE: Graph Level

INPUT: RT:binary

OUTPUT: Real value between 0 and $(|R|-1) \times |T|$

Redundancy, Row

The mean number of row node edges in excess of one.

Netstat

TYPE: Graph Level

INPUT: N of dimension $m \times n$

OUTPUT: Real value between 0 and $(N-1) \times M$

Shared Situation Awareness

The similarity of actor pairs based on social interaction, physical distance, and socio-demographic data.

Graham, 2005

TYPE: Agent Level

INPUT: AA: interaction/communication, AA: physical proximity, AA: socio-demographic similarity

OUTPUT: Real value between 0 and $(N-1) \times M$

Simmelian Ties

Computes the normalized number of nodes to which each node has a Simmelian tie.

Krackhardt, 1998

TYPE: Node Level

INPUT: N: square

OUTPUT: Real value between 0 and 1

Span of Control

The average number of out edges per node with non-zero out degrees.

Carley, 2002

TYPE: Graph Level

INPUT: N:square

OUTPUT: Real value between 0 and $|V| - 1$

Speed, Average

The average shortest path length between node pairs (i,j) where there is a path in the network from i to j . If there are no such pairs, then Average Speed is zero.

Carley, 2002

TYPE: Graph Level

INPUT: N:square

OUTPUT: Real value between 0 and 1

Speed, Minimum

The maximum shortest path length between node pairs (i,j) where there is a path in the network from i to j . If there are no such pairs, then Minimum Speed is zero.

Carley, 2002

TYPE: Graph Level

INPUT: AA

OUTPUT: Real value between 0 and 1

Task Completion, Knowledge Based

The percentage of tasks that can be completed by the agents assigned to them, based solely on whether the agents have the requisite knowledge to do the tasks.

Carley, 2002

TYPE: Graph Level

INPUT: AK:binary; AT:binary; KT:binary

OUTPUT: Real value between 0 and 1

Task Completion, Resource

The percentage of tasks that can be completed by the agents assigned to them, based solely on whether the agents have the requisite resource to do the tasks.

Carley, 2002

TYPE: Graph Level

INPUT: AK:binary; AT:binary; KT:binary

OUTPUT: Real value between 0 and 1

Task Completion, Overall

The percentage of tasks that can be completed by the agents assigned to them, based on whether the agents have the requisite knowledge and resources to do the tasks.

Carley, 2002

TYPE: Graph Level

INPUT: AK:binary; AT:binary; KT:binary

OUTPUT: Real value between 0 and 1

Task Completion, Overall

The percentage of tasks that can be completed by the agents assigned to them, based on whether the agents have the requisite knowledge and resources to do the tasks.

Carley, 2002

TYPE: Graph Level

INPUT: AK:binary; AT:binary; KT:binary

OUTPUT: Real value between 0 and 1

Transitivity

The percentage of edge pairs $\{(i,j), (j,k)\}$ in the network such that (i,k) is also an edge in the network.

NetStat

TYPE: Graph Level

INPUT: N:square

OUTPUT: Real value between 0 and 1

Triad Count

The number of triads centered at each node in a square network.

NetStat

TYPE: Agent Level

INPUT: N:square of dimension $|V|$

OUTPUT: Integer between 0, $(|V|-1)(|V|-2)$

Under Supply, Knowledge

The extent to which the knowledge needed to do tasks are unavailable in the entire organization.

Carley, 2002

TYPE: Graph Level

INPUT: AK:binary; AT:binary; KT:binary

OUTPUT: Real value between 0 and 1

Under Supply, Resource

The extent to which the knowledge needed to do tasks are unavailable in the entire organization.

Carley, 2002

TYPE: Graph Level

INPUT: AK:binary; AT:binary; KT:binary

OUTPUT: Real value between 0 and 1

Upper Boundedness, Krackhardt

The degree to which pairs of agents have a common ancestor.

Krackhardt, 1994

TYPE: Graph Level

INPUT: N:square

OUTPUT: Real value between 0 and 1

Tasks

This folder explains how to analyze your network or organization by taking you step-by-step through common tasks such as simplifying complex visual networks or how to create your own MetaMatrix from scratch.

[Creating A MetaMatrix From An Excel Spreadsheet](#)

[Running An Over-Time Analysis Using The Over-Time Viewer](#)

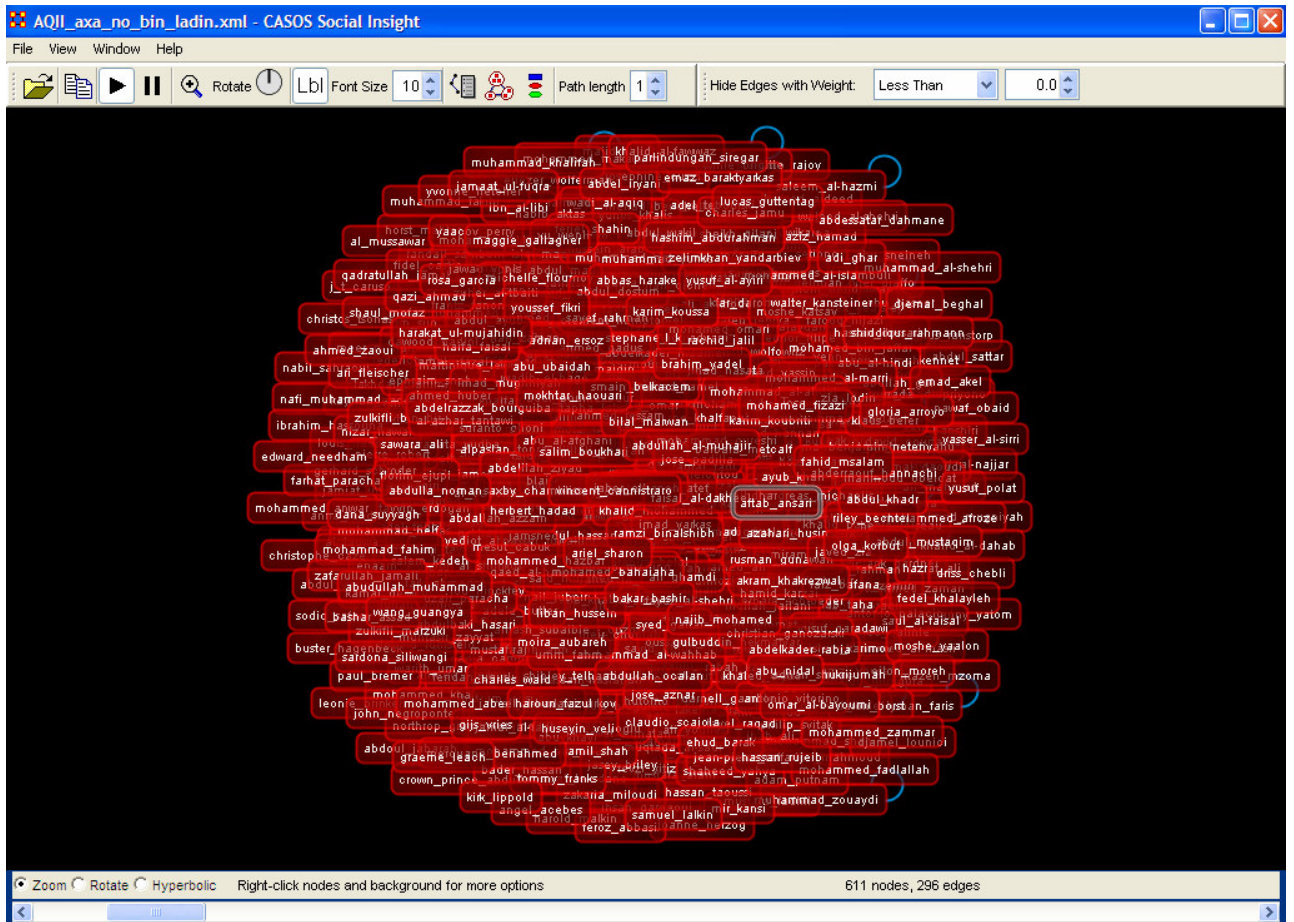
[Simplifying A Complex Visual Network](#)

[Optimizing Your Network](#)

Simplifying A Complex Visual Network

After you load a MetaMatrix, depending on its complexity, it may appear to be a jumbled ball of yarn (see screen shot below for an example of what this can look like); so, now what?

Tip! This help window best viewed maximized.



Far too dense and complex to be of practical comprehension, it is time to simplify this visualization. To do this you will need to become familiar with the Visualizer tool bar and learn how to interact with the visualization.

[Working Inside The Visualizer](#)

Working Inside The Visualizer

The next subset of help documents focus on simplifying a network from within the ORA Visualizer. Basic features of the Visualizer interface will be explained and common tasks explored. The goal is to become comfortable with the primary features of the Visualizer tool bar and window.

[The Tool Bar Explained](#)

[Eliminating Labels](#)

[Removing Isolates](#)

[Removing Pendants](#)

[Creating MetaNodes \(Group Nodes\)](#)

[Zooming](#)

[Hyperbolic View](#)

[Rotating A Visualization](#)

The Tool Bar Explained

ORA loads complex network data in stages to maximize your PC's efficiency, which is why when you first load a MetaMatrix it may appear highly condensed. Otherwise, it would take a while before anything appeared in the Visualizer.

When you first load a MetaMatrix you will see the "pause button" depressed in the Visualizer tool bar. You can "layout" the network by clicking the "play" button. The visualization will space apart across the screen. This is called "laying out" a MetaMatrix. The Visualizer is separating nodes and ties that at first appear on top of each other. You can "pause" this process again when you think the network is satisfactorily spaced apart or you can let ORA lay out the entire MetaMatrix. Your computer speed as well as the complexity of MetaMatrix will determine how long this takes.

Scroll down for a series of screen shots and explanations of the Visualizer tool bar.

Open Folder

The open folder icon allows you to load another MetaMatrix and works the same as the open folder icon in the main interface. The black circle below highlights where to find the open folder icon on the tool bar.



Copy Paste

The Copy Paste feature allows you quickly capture a visualization and save it to be pasted into another document later. The black circle below highlights how to access the Copy Paste feature from the Visualizer tool bar.



Play / Pause

The Play Pause function works very similar to the play pause feature on any device. The pause button (two vertical bars) is depressed, the Visualizer stops laying out a network. When the play button is depressed (right pointed triangle) ORA Visualizer begins laying out the network as described above. The black ellipse below highlights on the tool bar where this feature is located and accessible.



Magnifying / Maximizing

The magnifying glass icon with the plus sign inside it, allows you to instantly fill the Visualizer window pane with the currently rendered MetaMatrix. The black circle in the screen shot below highlights where to access this feature on the Visualizer tool bar.

Tip! This function works well when selecting a small part of your overall network and magnifying it to fill the Visualizer screen.



Rotating The Visualization

To rotate your visualization look for the word next to the Noon-pointing sundial icon in the tool bar. This feature is another way to manipulate your visualization to fill the entire Visualizer window pane.

Tip! Use this feature in conjunction with the magnifying glass function to work your visualization into largest size possible.

The black ellipses below highlight where this feature can be accessed on the Visualizer tool bar.



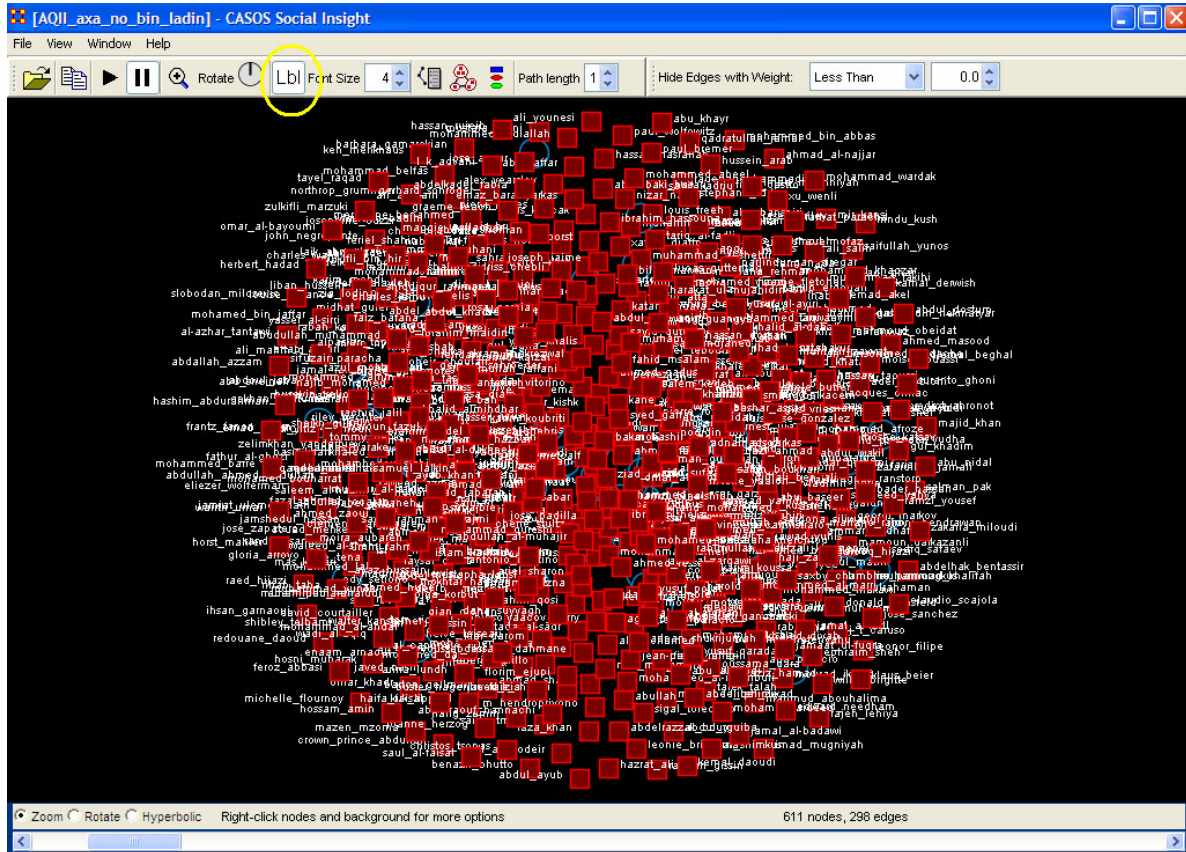
Eliminating Labels

Node labels, which often prove helpful in describing node points, can sometimes clutter a visualization. Thus, it may be necessary to eliminate labels from your visualization. In the screen shot below, the clarity of the network is clouded by the many labels.

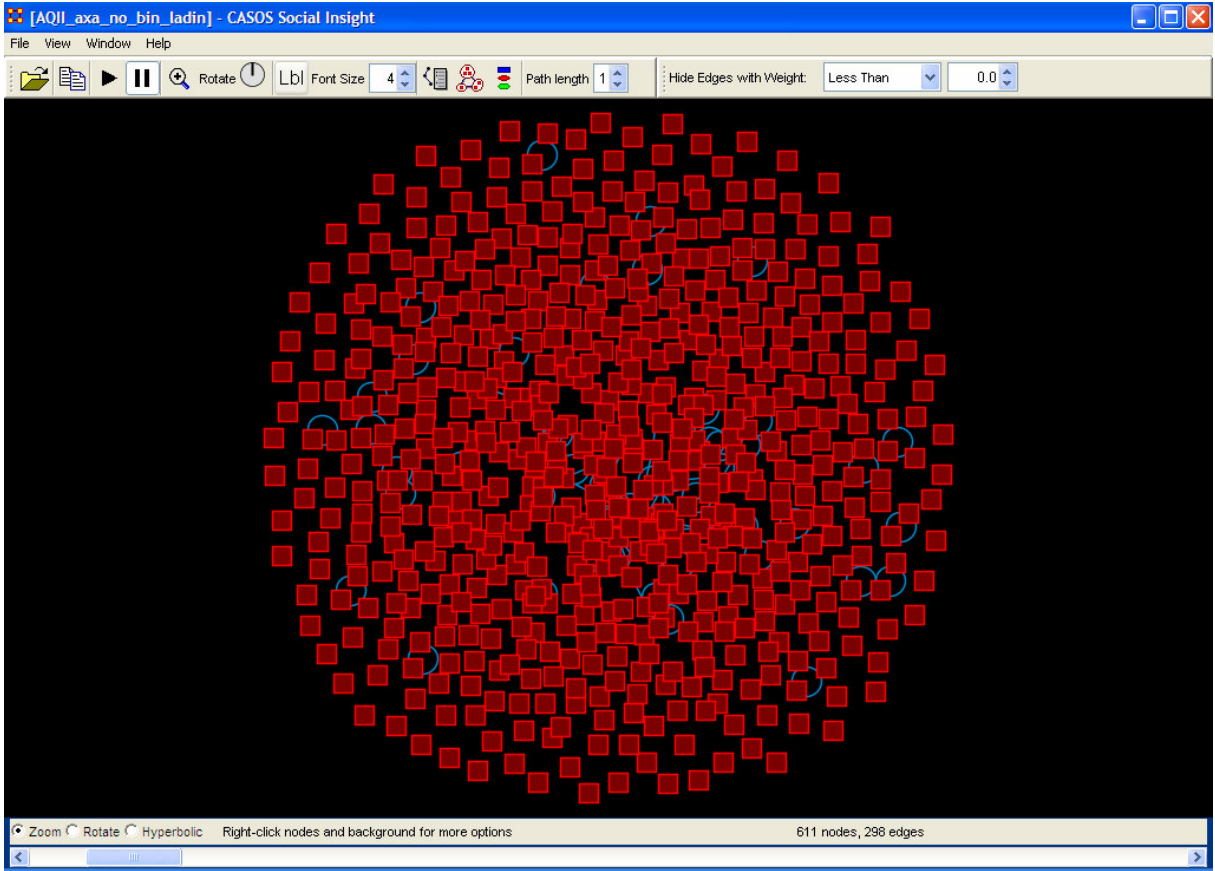
To remove labels > go to Visualizer tool bar > click the depressed label button

The yellow ellipse highlights where to find the Label Button on the Visualizer tool bar, which is abbreviated "Lbl."

(Scroll down below this screen shot to see another, where the labels have been removed)

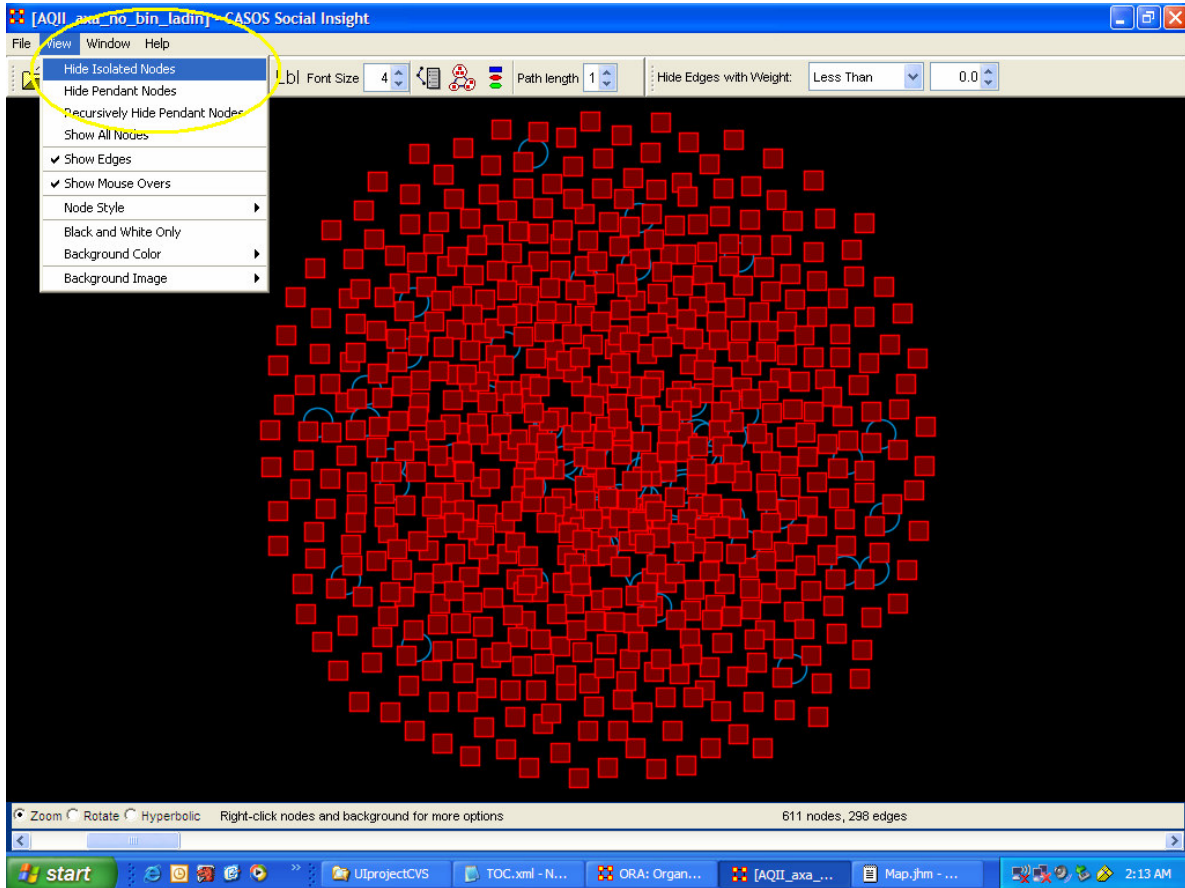


When you click the Label button on the Visualizer tool bar, which is depressed by default, ORA removes all labels. In the screen shot below, our example above has the labels removed. Note that now a much more clearer conceptual picture of the network is produced.

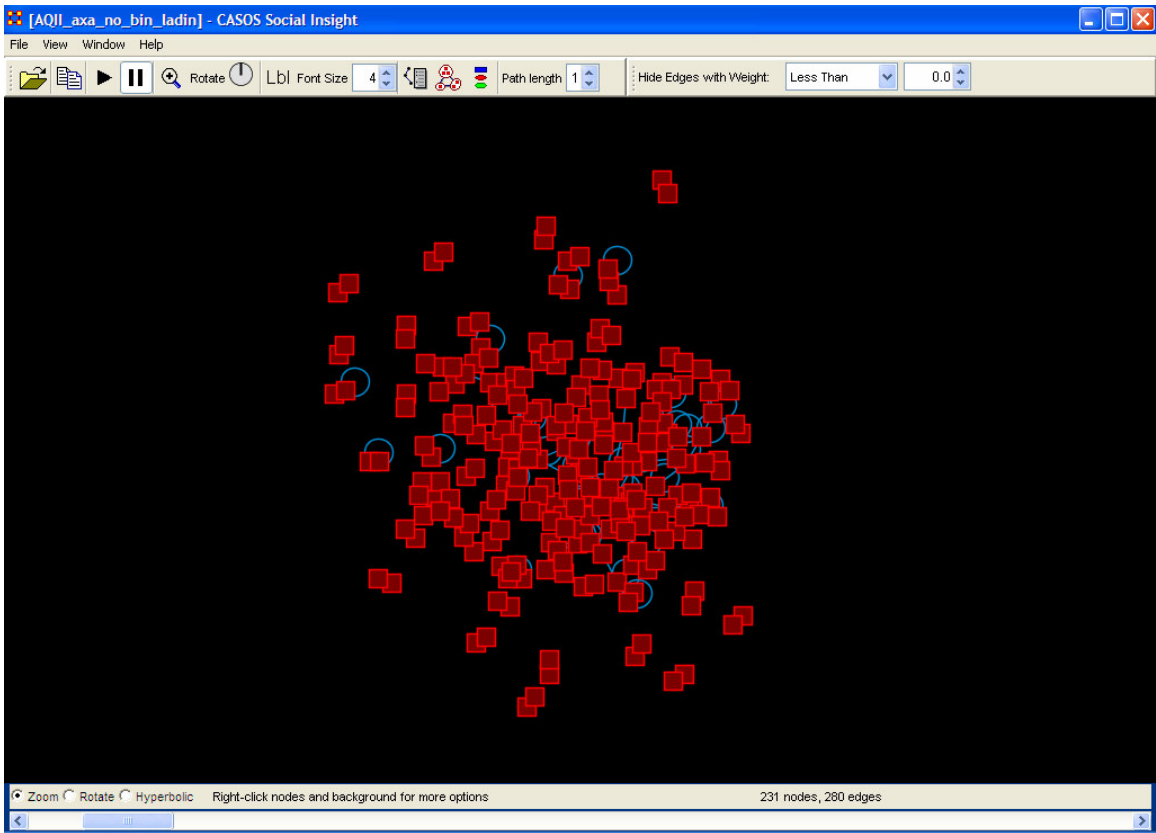


Removing Isolates

To further simplify a visualization, it may prove useful to remove isolated nodes from the visualization. Isolated nodes are nodes not directly linked or connected to other nodes, which share direct ties with each other. In the screen shot below, the yellow ellipse highlights how to access the "Remove Isolate Node" function. Agents without connections to other Agents will be removed. Scroll down for an example of the below visualization with the Isolated nodes removed.



In the screen shot below, after removing isolated nodes, the visualization is further simplified.

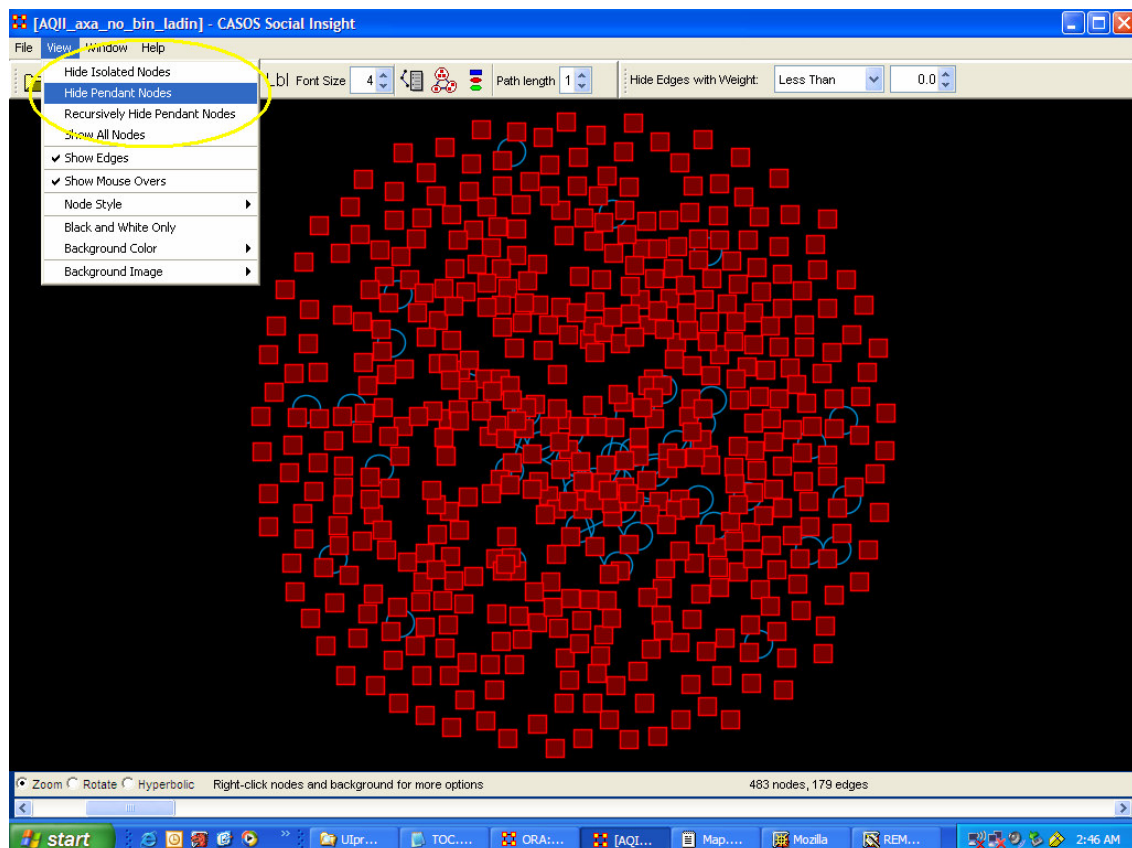


Removing Pendants

Pendant nodes share links to other nodes but their linkage is tangential and therefore isolated from the core linkages you may be interested in examining. They too, like isolated nodes, can be removed.

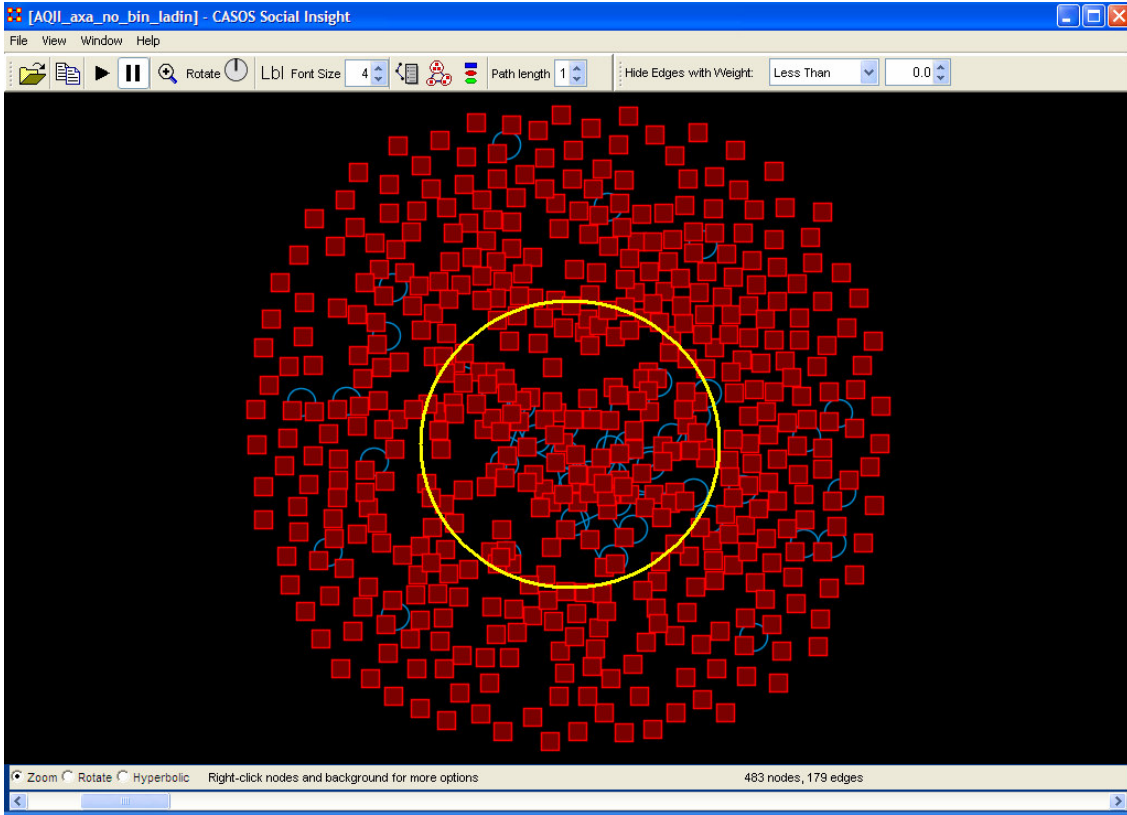
From the drop down menu in the Visualizer tool bar > View > Hide Pendant Nodes

The yellow ellipse in the screen shot below, shows where to access the Hide Pendant Nodes function. Scroll down below this image for a screen shot of pendant nodes removed.



Below is a screen shot of the above visualization after pendant nodes had been removed. The yellow ellipse highlights the area of the visualization that is less dense than in the screen shot above. This is a result of removing the pendant nodes.

Tip! You may have to compare the screen shots carefully.

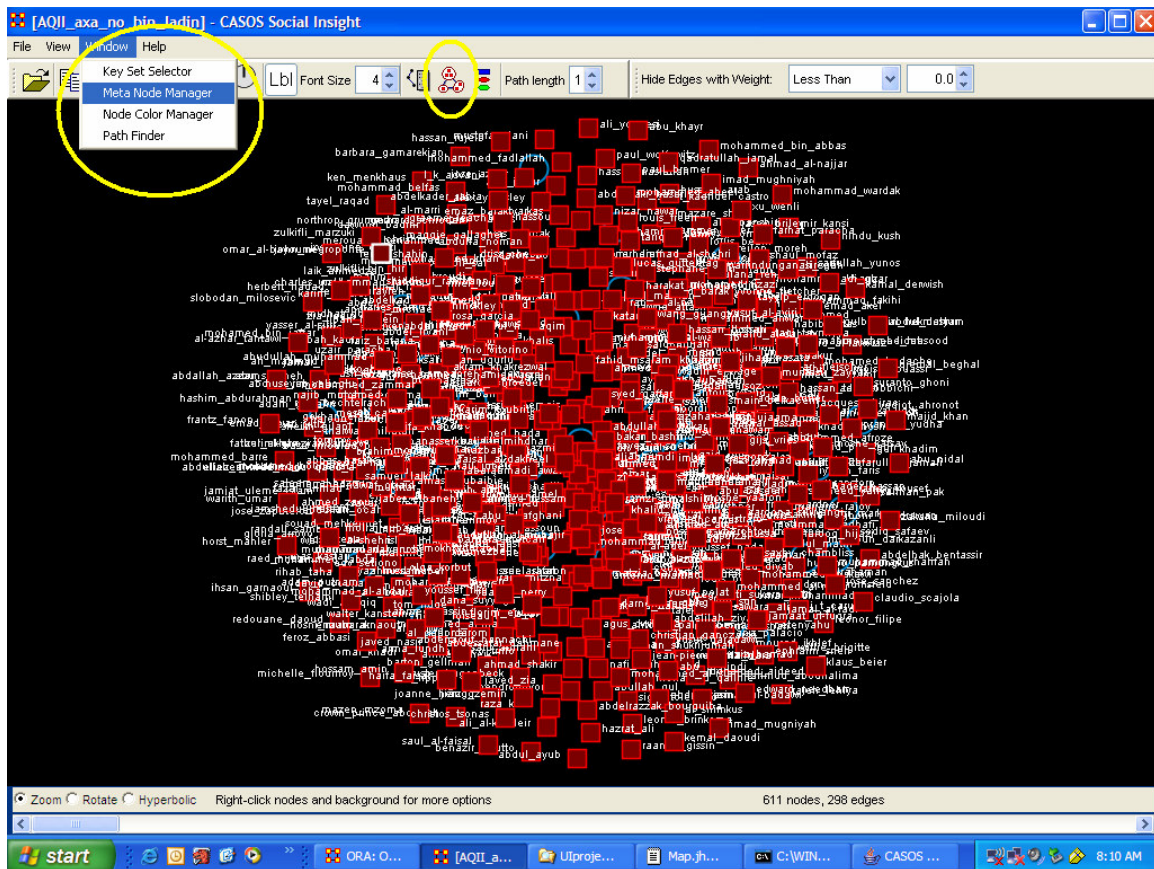


Creating A MetaNode

A *MetaNode* contains multiple nodes collapsed into one. You can create MetaNodes based on the nodesets in your organization, or you can create MetaNodes based on the attributes of the nodes. To create MetaNodes, you must access the *MetaNode Manager*. There are two ways to do this task:

1. From the drop down menu > Window > MetaNode Manager
2. from the Visualizer tool bar

The yellow ellipses below highlight where to access the MetaNode Manager through the drop down menu and the Visualizer tool bar.

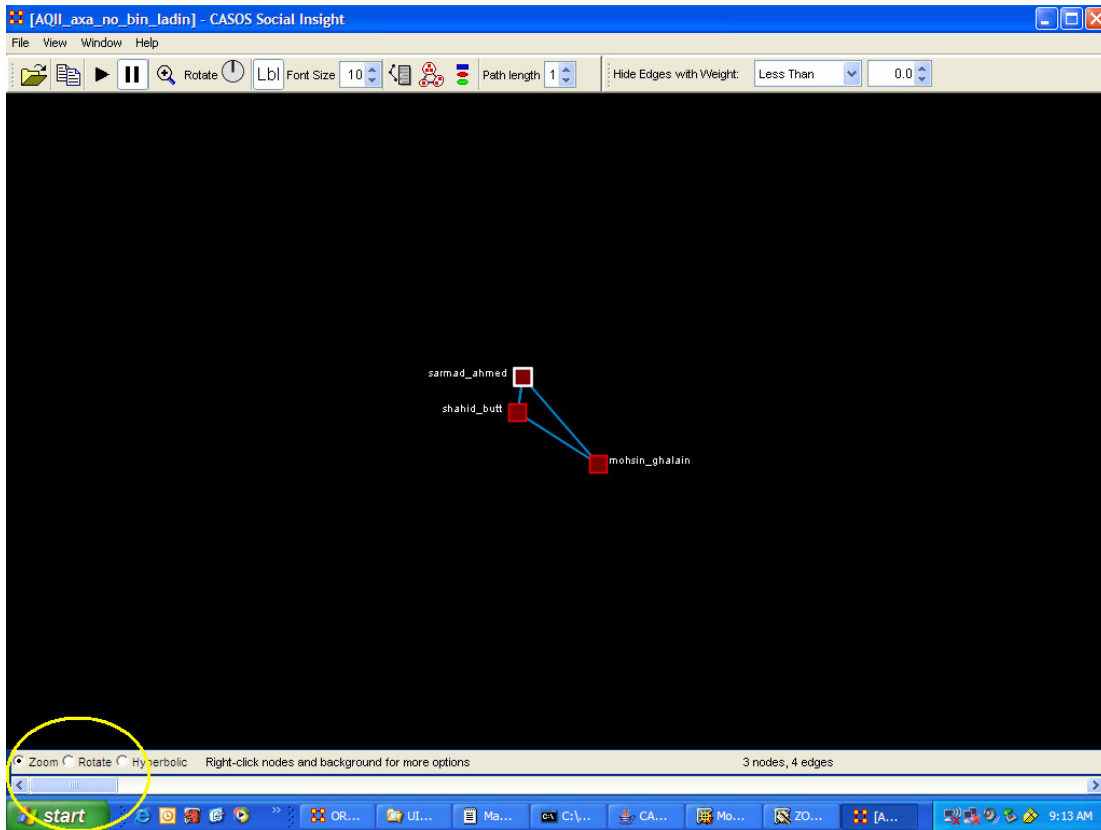


In the screen shot below, the yellow ellipses highlight how to create MetaNodes based on *Attribute 1* of our *Agent by Agent MetaMatrix* graph. The Visualizer itself shows your condensed visualization. All the nodes, which share the same attributes, are now groped into MetaNodes.

To view all the original nodes, click expand all MetaNodes. To create additional MetaNodes, click Create MetaNodes and select another attribute. You can only create

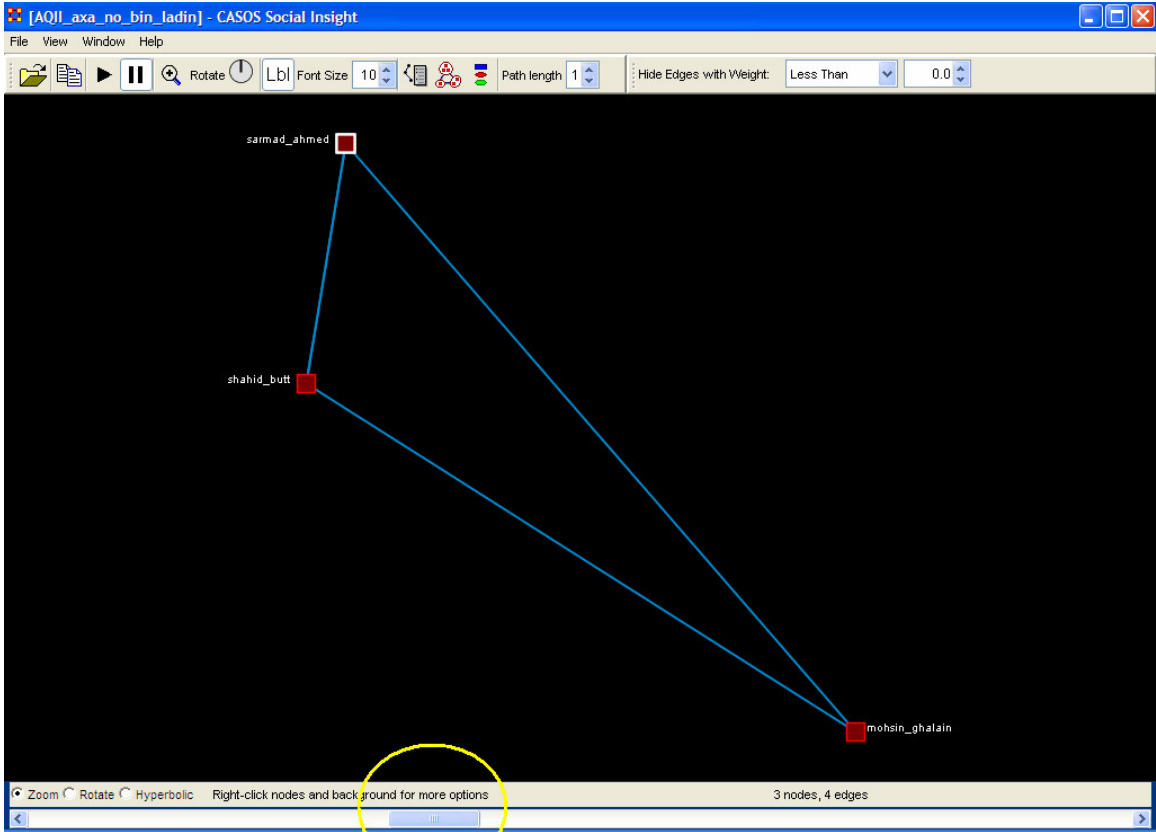
Zooming

At the bottom of the Visualizer interface is a sliding zoom bar. The yellow ellipse below highlights this feature. You can drag the slider it to the left or to the right increasing or decreasing the magnification of your visualization. In this example, we will click on the slider and move it toward the right. Scroll down to see a screen shot of the following visualization magnified.



In the screen shot below, note the position of the slider on the slide bar highlighted with the yellow ellipse. You can increase the magnification to the point that parts of the visualization will move off the screen.

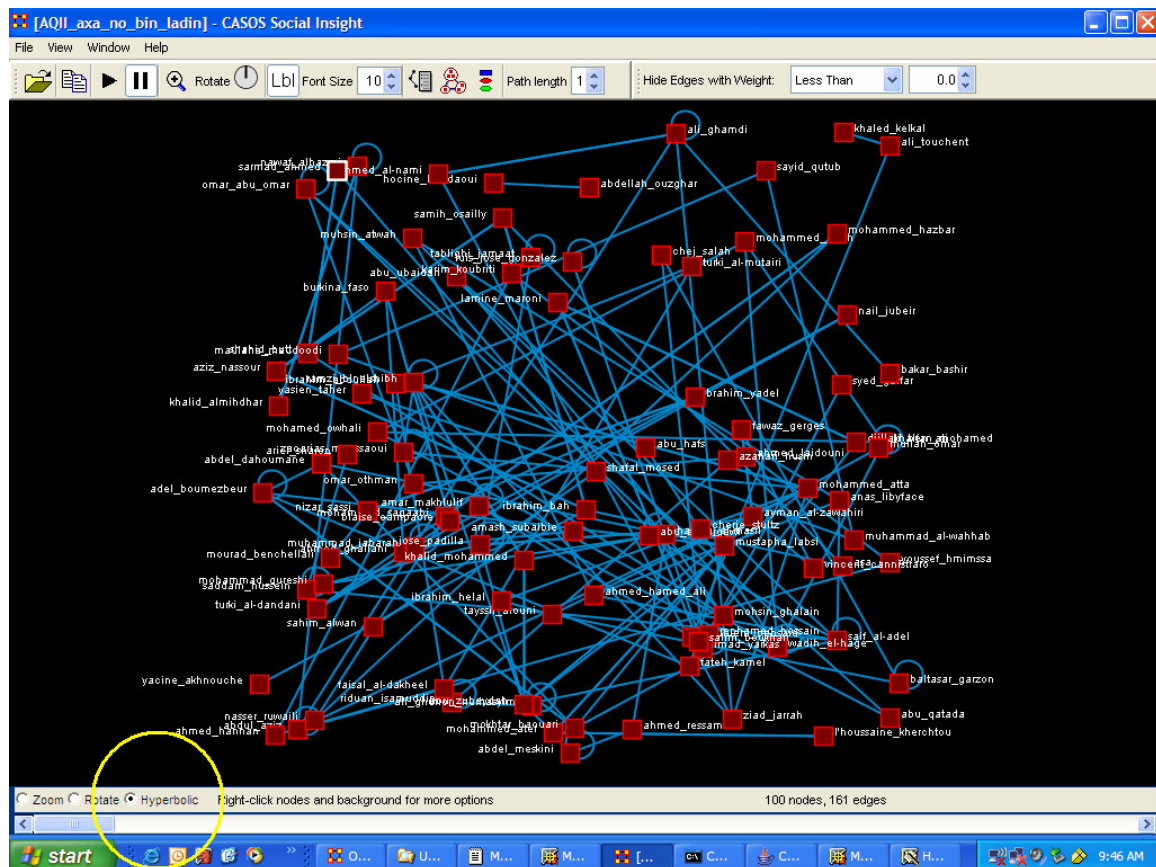
Tip! Use the Magnifying glass in the Visualizer tool bar to zoom your visualization to the greatest possible size, which still captures the entire visualization.



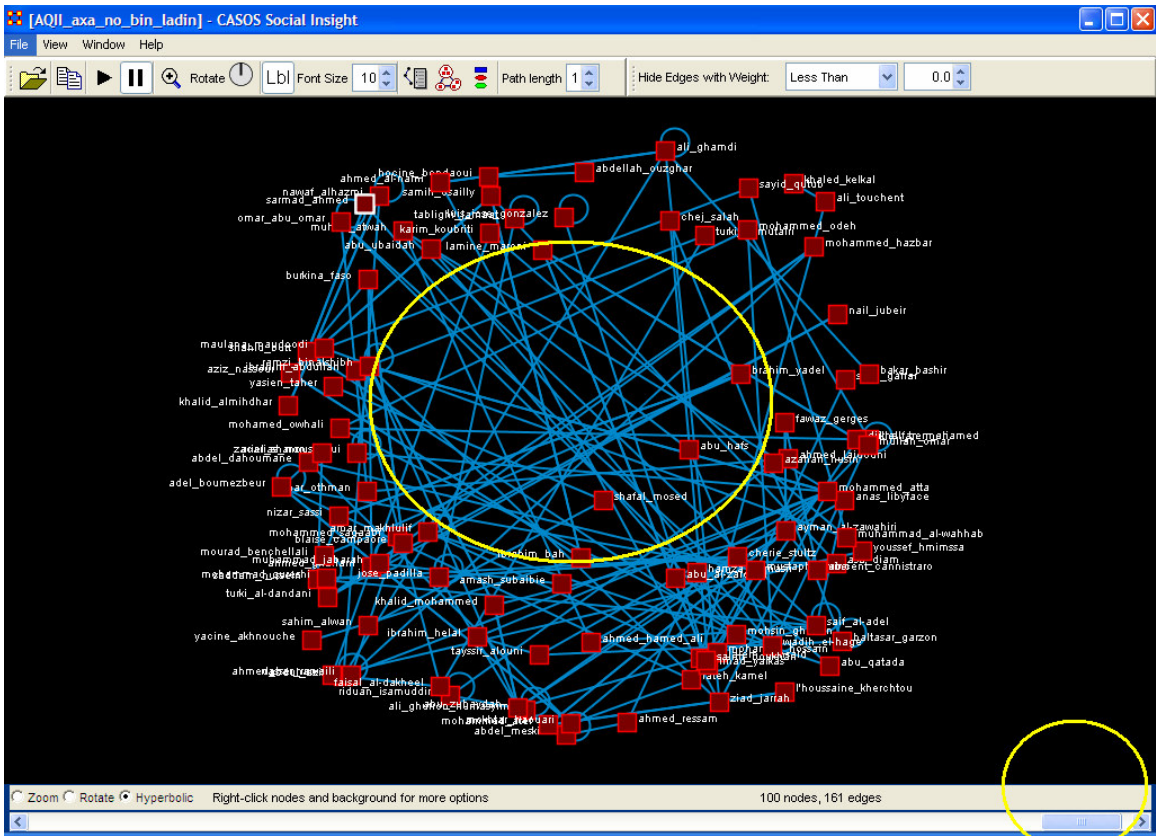
Hyperbolic View

The Hyperbolic function creates a "bulge" within your visualization adding a sense of depth. By moving the slider from right to left, you can increase or decrease this bulge effect and create different depth-added views. The yellow circle in the screen shot below highlights where to access the Hyperbolic function.

Tip! The Hyperbolic function, when selected, converts the slider to this feature. This is also true of using the zoom and rotate functions. Use all three in conjunction to get your visualization just the way you want it. Scroll down below this screen shot for another when viewed with the Hyperbolic feature.



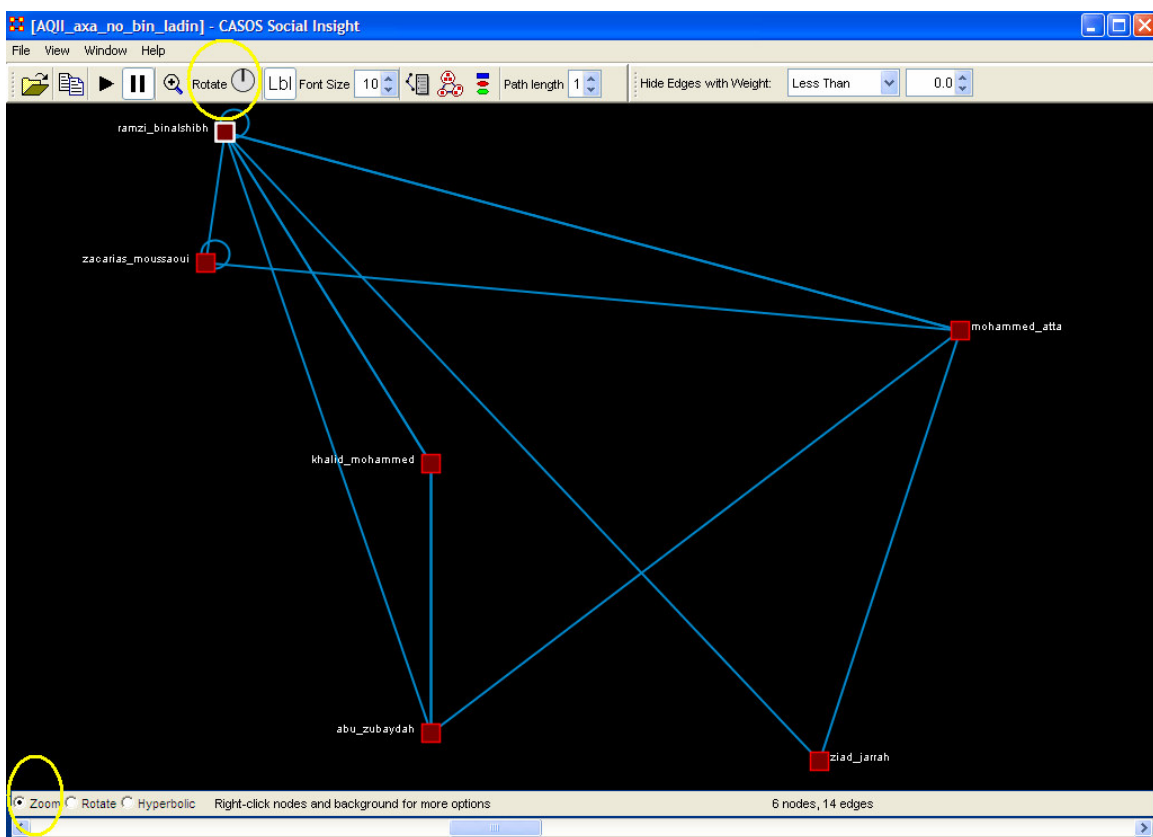
In the screen shot below, yellow ellipses highlight both the effect of adding the Hyperbolic View to a visualization as well as the slider position that created it. Compare the slider positions and visualizations in both screen shots to examine this subtle effect. Of course, the best way to become familiar with the Hyperbolic view is to simply experiment within the Visualizer itself.



Rotating A Visualization

You can rotate a visualization with the slider bar at the bottom of the interface or from the tool bar at the top. To do this from the slide bar, click the rotate option toward the bottom of the interface. This activates the rotate feature. As you move the slider from the left to right, your visualization will rotate correspondingly. From the tool bar, you can click inside the rotate icon, which will turn the visualization depending on exactly where you clicked. In the screen shot below, the yellow ellipses highlight where to access ORA's rotate tool both at the bottom and top of the visualizer interface.

Tip! Use the rotate feature in tandem with the magnifying glass to find a visualization's maximum screen size.



Adding And Removing Nodes

This subfolder contains instruction on Basic methods to add nodes and remove them from your MetaMatrix. Adding and removing nodes in the MetaMatrix Editor, Visualizer and Key Set will be explained. Advanced usages will include more thorough guidance on adding and removing nodes.

[In The Editor](#)

[In The Visualizer](#)

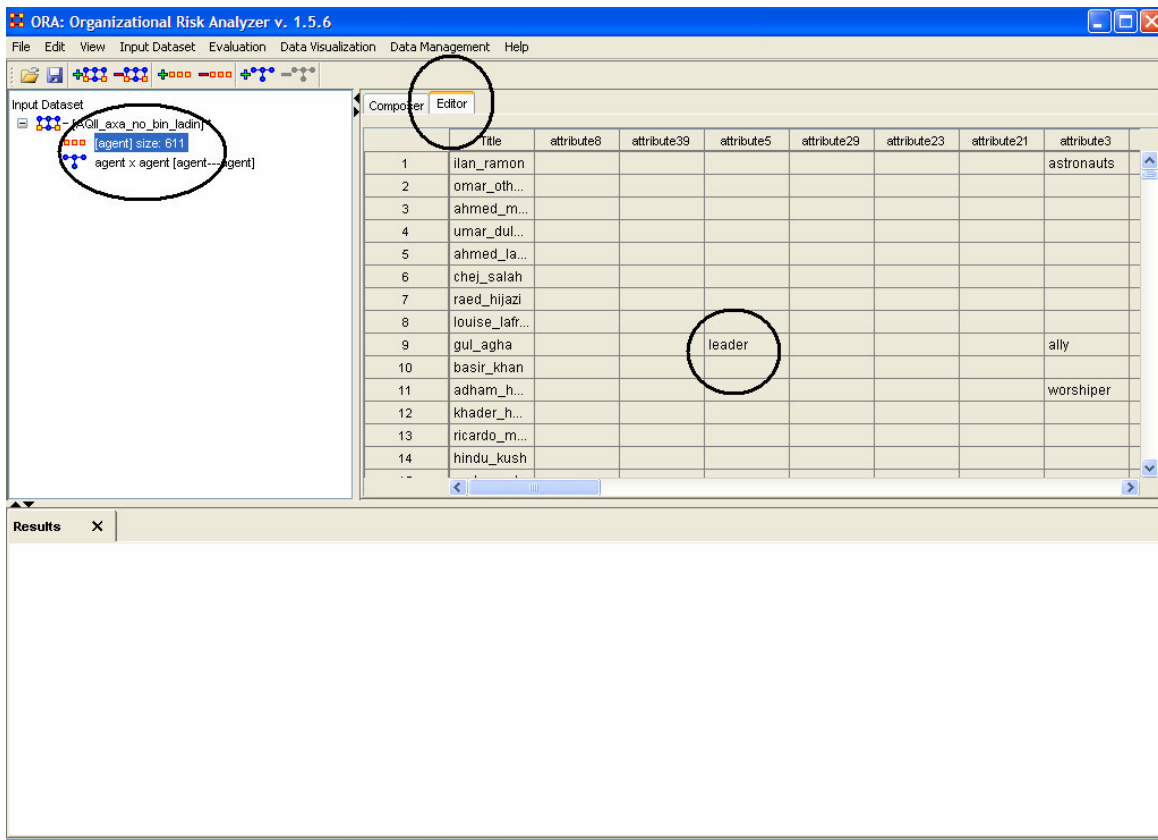
[In The Key Set](#)

Adding and Removing Nodes In The Editor

The MetaMatrix Composer is accessed through ORA's main interface and allows you to add or remove nodes and nodesets by creating or removing "attributes" you wish to associate with your MetaMatrix. An attribute is a label that you wish to give a particular node and can literally be anything you wish (e.g., worker, leader, Africa, March, 1989, etc.) Attributes can be used to define nodes and nodesets according to your specific descriptions. For example, if you wanted to add the attribute "leader" to describe several nodes, you can go directly inside the MetaMatrix Editor and give attributes to those nodes which you attribute the description "leader."

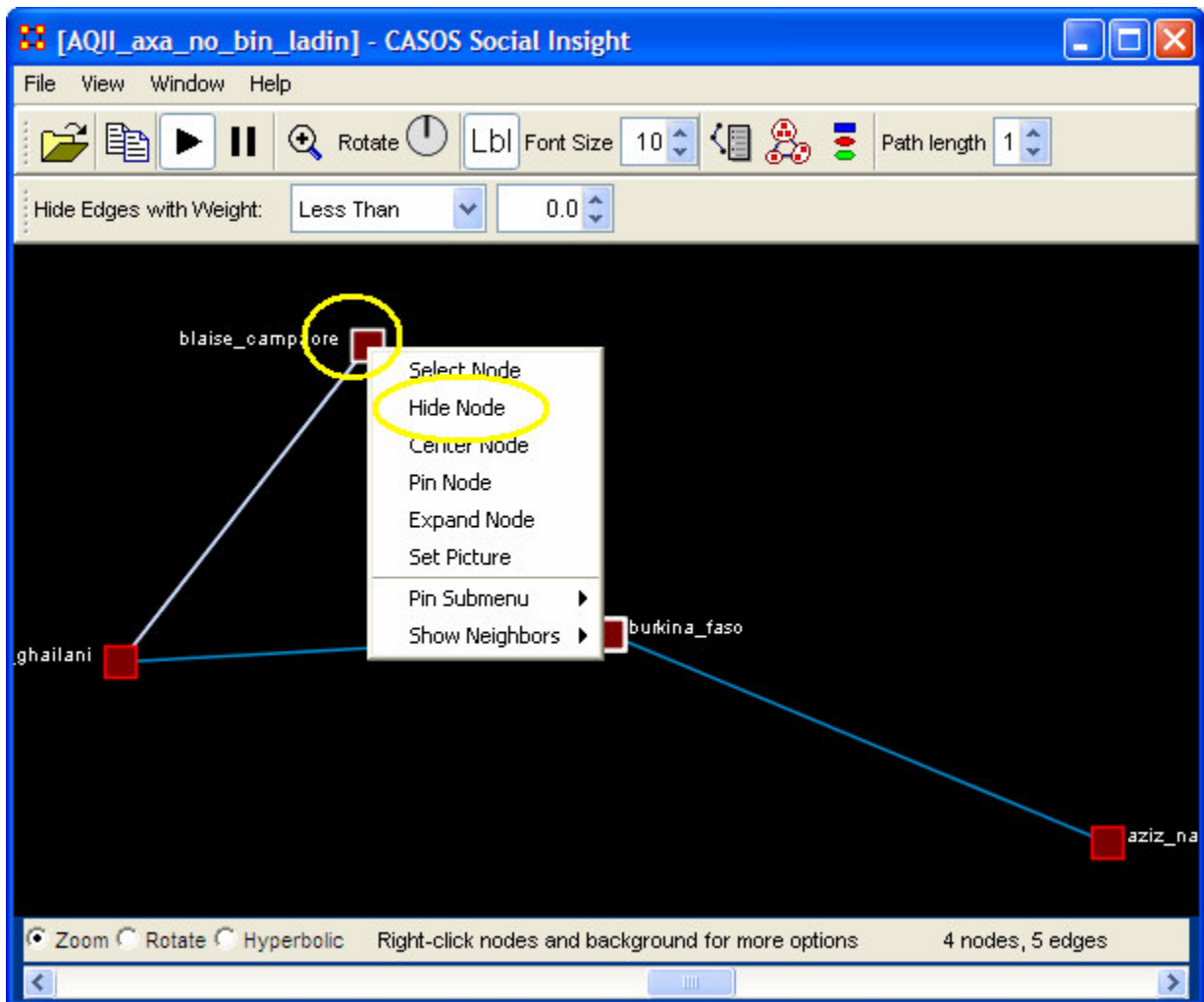
From ORA's main interface > "Editor" tab > Enter desired attribute under selected row and column for appropriate node.

The black ellipses below highlight how to access the MetaMatrix Editor from ORA's main interface.

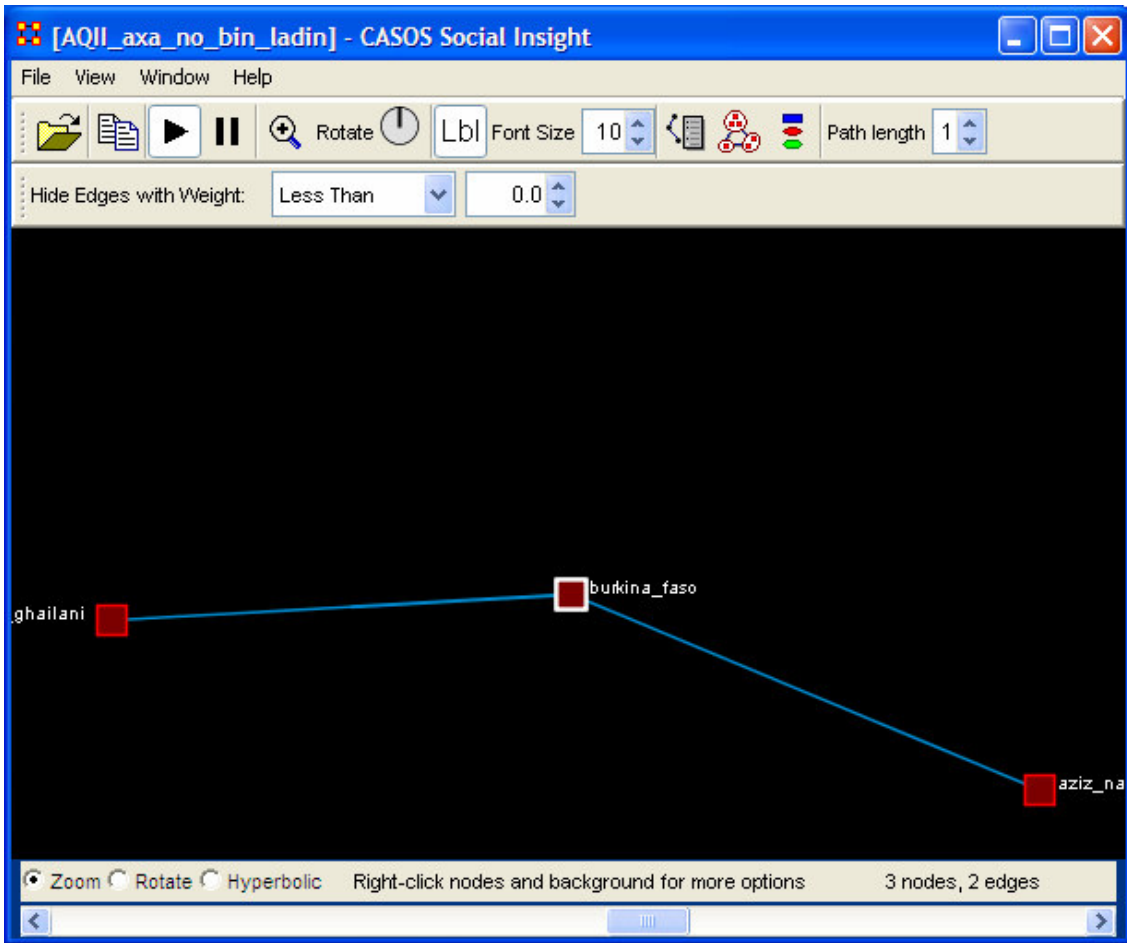


Adding and Removing Nodes In The Visualizer

You can remove nodes directly from the ORA Visualizer by right clicking on a node and selecting "Hide Node." You can reverse this process by clicking "expand nodes." The yellow ellipses below highlight how to add and remove a node from inside the ORA Visualizer. Scroll down below for a screen shot of the following visualization with the selected node removed.



Below is screen shot of the above visualization with the "blaise_campaore" node removed.

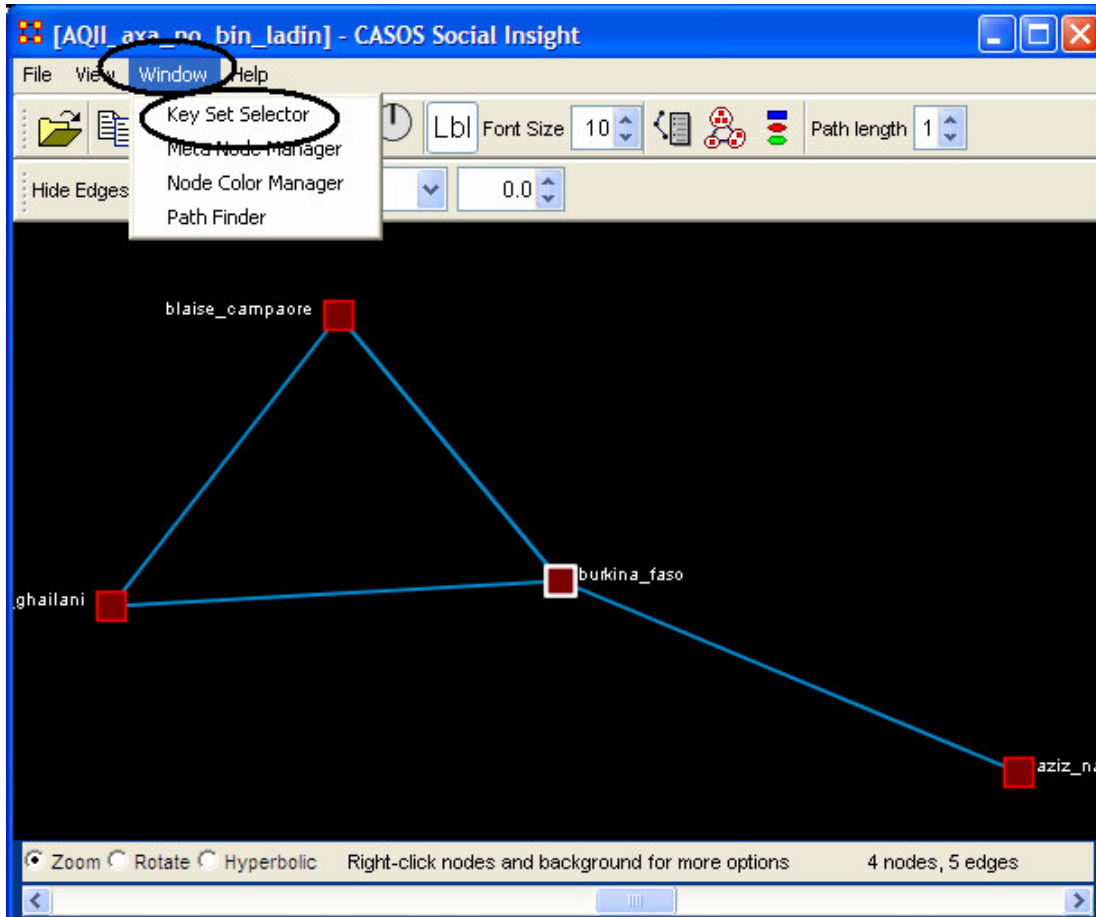


Adding and Removing Nodes in the Key Set

The ORA Key Set Selector allows you to determine which nodes are visible in the Visualizer. The Key Set Selector is accessible through the ORA tool bar.

From the ORA Tool Bar > Window > Key Set Selector

The black ellipses below show where to access the ORA Key Set Selector (Scroll down for a screen shot of the Key Set Selector window pane and additional instruction).

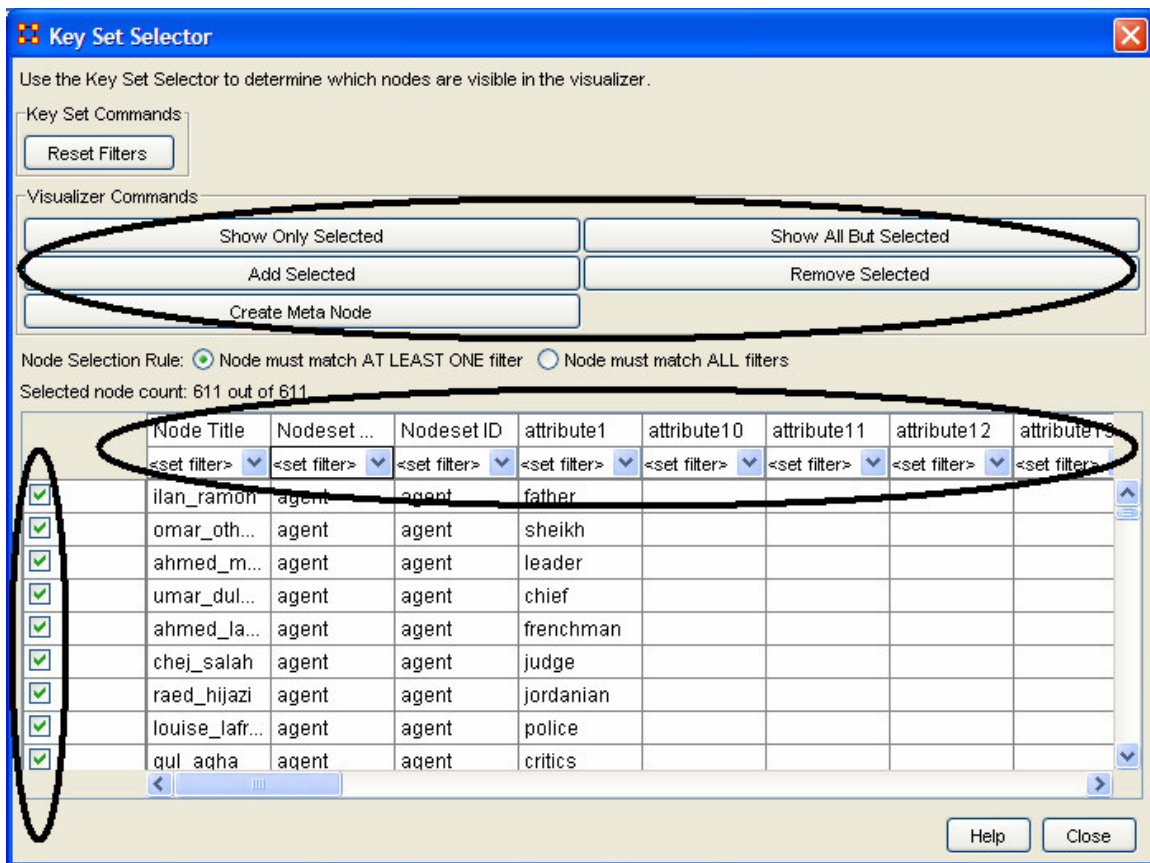


Below is a screen shot of the Key Set Selector accessible through the ORA Visualizer tool bar. The black ellipses below highlight the primary areas of the Key Set Selector, which enable you to hide and add nodes to your visualization. Note the green checkmarks along the first column. These checkmarks indicate exactly which nodes within your visualization are "selected." The button within the Visualizer Commands section of the window pane, allow you to view your visualization by "Show Only Selected" or "Show All But Selected."

Furthermore, Visualizer Commands allows you to "Add Selected," "Remove Selected," and "Create Meta Nodes." Note the difference between "Showing" and "Adding" and

conversely "Show All But Selected" and "Remove Selected." Essentially, when you "add" you are literally adding nodes to your visualization versus showing, which is displaying nodes already part of the current visualization.

The Node Title Row, highlighted by the middle black ellipse on the screen shot below, allows you "filter" nodes in your visualization. To view only the nodes in your visualization that contain a certain attribute, which you are interested in examining, you can turn the filter "on" by clicking the drop down tab in the row immediately following the Node Title row (Scroll for another screen shot of filtering the current MetaMatrix using "attribute 1" from the Key Set Selector column heading).



In the screen shot below, note that using the filters we selected under "attribute 1" the attribute "1988" from the drop down menu. A black ellipse highlights that by selecting this attribute, "4 out of 611" nodes contain that attribute. By clicking "Show Only Selected" the ORA Visualizer will render these four nodes. Note that the green check marks in the first column have been unselected. Only the nodes that shared the 1988 attribute will be selected. If you scroll down this column, those nodes will turn up as selected. By clicking "Show All But Selected" the ORA Visualizer will render the MetaMatrix without the four nodes containing the attribute "1988." Other filters can be used to create a variety of visualizations correlating any attributes that you are interested in examining. Note also that you will note be able to add attributes directly into the Key Set Selector. This must be done in the MetaMatrix Editor (See Adding and Removing

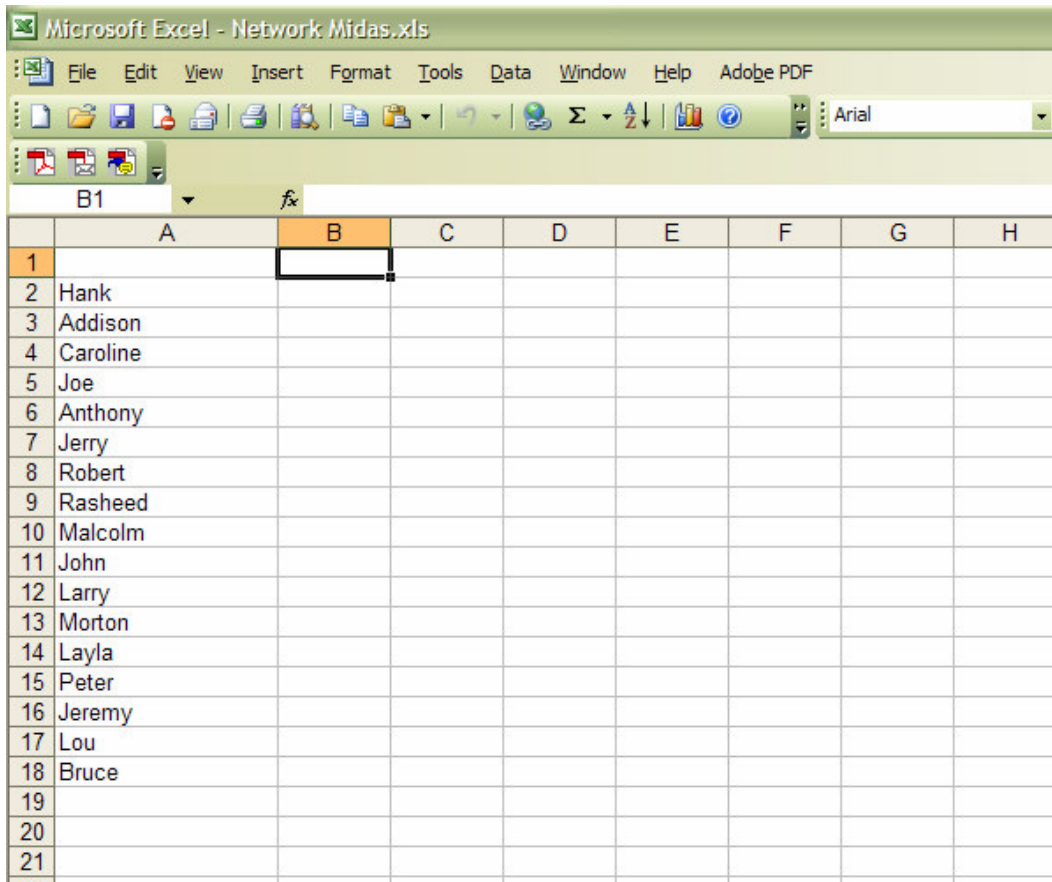
Creating a MetaMatrix from an Excel Spreadsheet

If you don't have a MetaMatrix, you can create one from scratch. Below is step-by-step instruction on how to do this.

We will create "Network Midas" an agent-by-agent square MetaMatrix. We say it is "square" because all row headings correspond directly to column headings. This is important as it relates to specific measures ORA can run on a graph. If the graph is not square, some measures will not work (See [Measures](#) for additional instruction).

Begin by opening a blank Microsoft Excel work book. In "column A" we will enter the name of all agents that make up our social network or organization.

Note: When creating your spreadsheet, do not add any additional titles, notes, or other headings, which will interfere with the "square" properties of the MetaMatrix.



	A	B	C	D	E	F	G	H
1								
2	Hank							
3	Addison							
4	Caroline							
5	Joe							
6	Anthony							
7	Jerry							
8	Robert							
9	Rasheed							
10	Malcolm							
11	John							
12	Larry							
13	Morton							
14	Layla							
15	Peter							
16	Jeremy							
17	Lou							
18	Bruce							
19								
20								
21								

Next, create column headings using the correlating names as they appear in row headings. Again, this will ensure that our MetaMatrix will be square.

	A	B	C	D	E	F	G	H
1		Hank	Addison	Caroline	Joe	Anthony	Jerry	Robert
2	Hank							
3	Addison							
4	Caroline							
5	Joe							
6	Anthony							
7	Jerry							
8	Robert							
9	Rasheed							
10	Malcolm							
11	John							
12	Larry							
13	Morton							
14	Layla							
15	Peter							
16	Jeremy							
17	Lou							
18	Bruce							
19								
20								
21								

Here we will create "links (aka edges, and ties)" between each agent. We do this by entering a "1" if a direct connection or relationship exists and a "0" if it does not. Please note that headings that cross-reference themselves are considered redundant and thus are left blank or "0." For instance, in cell "B2" it is assumed "agent Hank" has a tie to himself.

In the example below, "Red Xs" are used to illustrate the redundant ties. This redundancy should continue as a smooth diagonal line from the top left corner of your MetaMatrix to the bottom right.

	A	B	C	D	E	F	G	H
1		Hank	Addison	Caroline	Joe	Anthony	Jerry	Robert
2	Hank							
3	Addison							
4	Caroline							
5	Joe							
6	Anthony							
7	Jerry							
8	Robert							
9	Rasheed							
10	Malcolm							
11	John							
12	Larry							
13	Morton							
14	Layla							
15	Peter							
16	Jeremy							
17	Lou							
18	Bruce							
19								
20								
21								

Tip! If you can't make this smooth diagonal line, your graph is not square.

Using 1s and 0s to establish linkages, complete your spreadsheet.

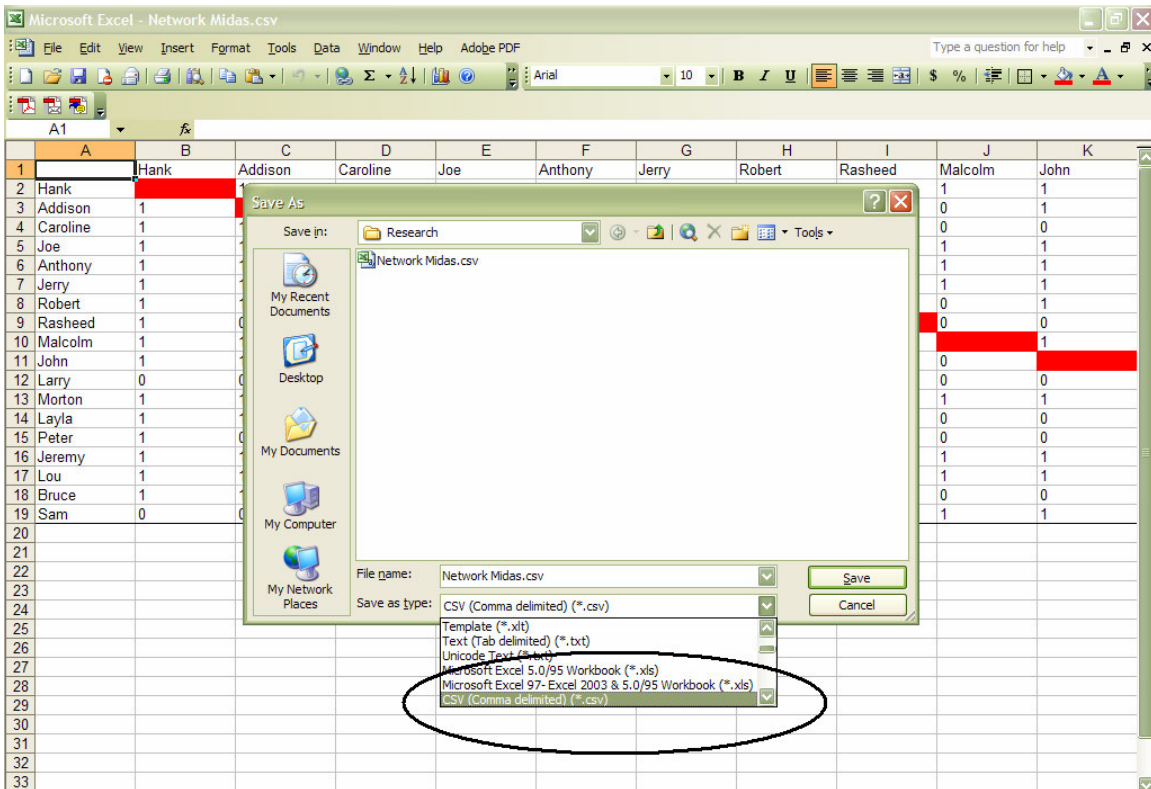
In the "Network Midas" example, we have assigned relationships randomly. Within your organization or network, however, you can describe any direct connections or relationships you are interested in analyzing. For instance, you may determine that a "direct connection" exists if agents within your network consult with each other at least once a month; literally, it can be anything you decide.

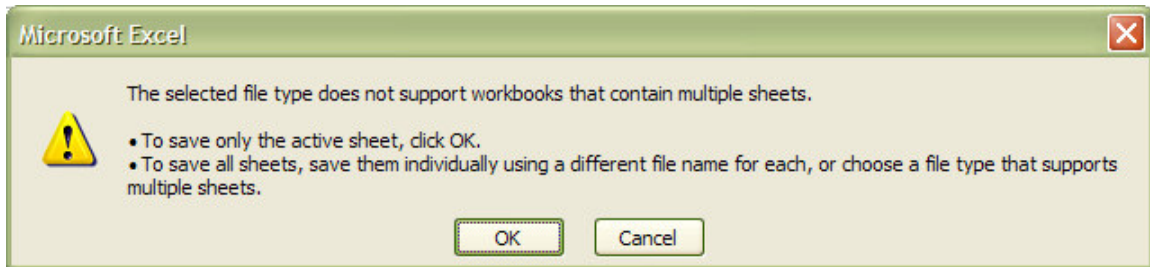
Below is our completed MetaMatrix, Network Midas (The red fill illustrates cells that do not require input due to their redundancy).

	A	B	C	D	E	F	G	H	I	J	K
1		Hank	Addison	Caroline	Joe	Anthony	Jerry	Robert	Rasheed	Malcolm	John
2	Hank	1	1	1	1	1	1	1	1	1	1
3	Addison	1	1	1	1	1	1	1	0	0	1
4	Caroline	1	1	1	1	0	1	1	0	0	0
5	Joe	1	1	1	1	1	1	1	0	1	1
6	Anthony	1	1	0	1	1	1	1	1	1	1
7	Jerry	1	1	1	1	1	1	1	0	1	1
8	Robert	1	1	1	1	1	1	1	0	0	1
9	Rasheed	1	0	0	0	0	1	0	1	0	0
10	Malcolm	1	1	0	1	1	1	1	1	1	1
11	John	1	1	0	0	0	0	0	0	0	1
12	Larry	0	0	0	0	0	0	0	0	0	0
13	Morton	1	1	0	1	1	1	1	1	1	1
14	Layla	1	1	1	1	0	1	1	1	0	0
15	Peter	1	0	0	0	0	0	0	0	0	0
16	Jeremy	1	1	1	1	0	1	1	0	1	1
17	Lou	1	1	1	0	0	1	1	0	1	1
18	Bruce	1	1	0	1	0	1	0	0	0	0
19	Sam	0	0	0	0	1	1	0	0	1	1

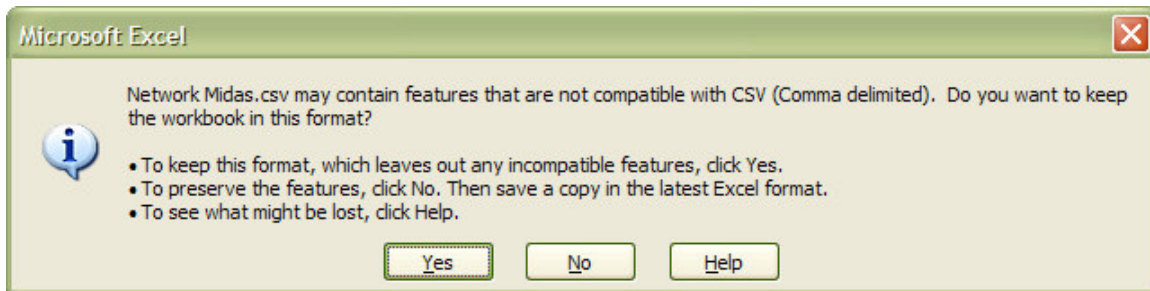
Now that we have essentially built a MetaMatrix from scratch using Excel, the next step is to save it in a compatible file format ORA can interpret. For Excel spreadsheets, in all likelihood, this will be the "CSV" (comma separate values) file format.

From the drop down menu: File > Save As > Save As Type > CSV (comma delimited) (*.csv)





Click "OK"

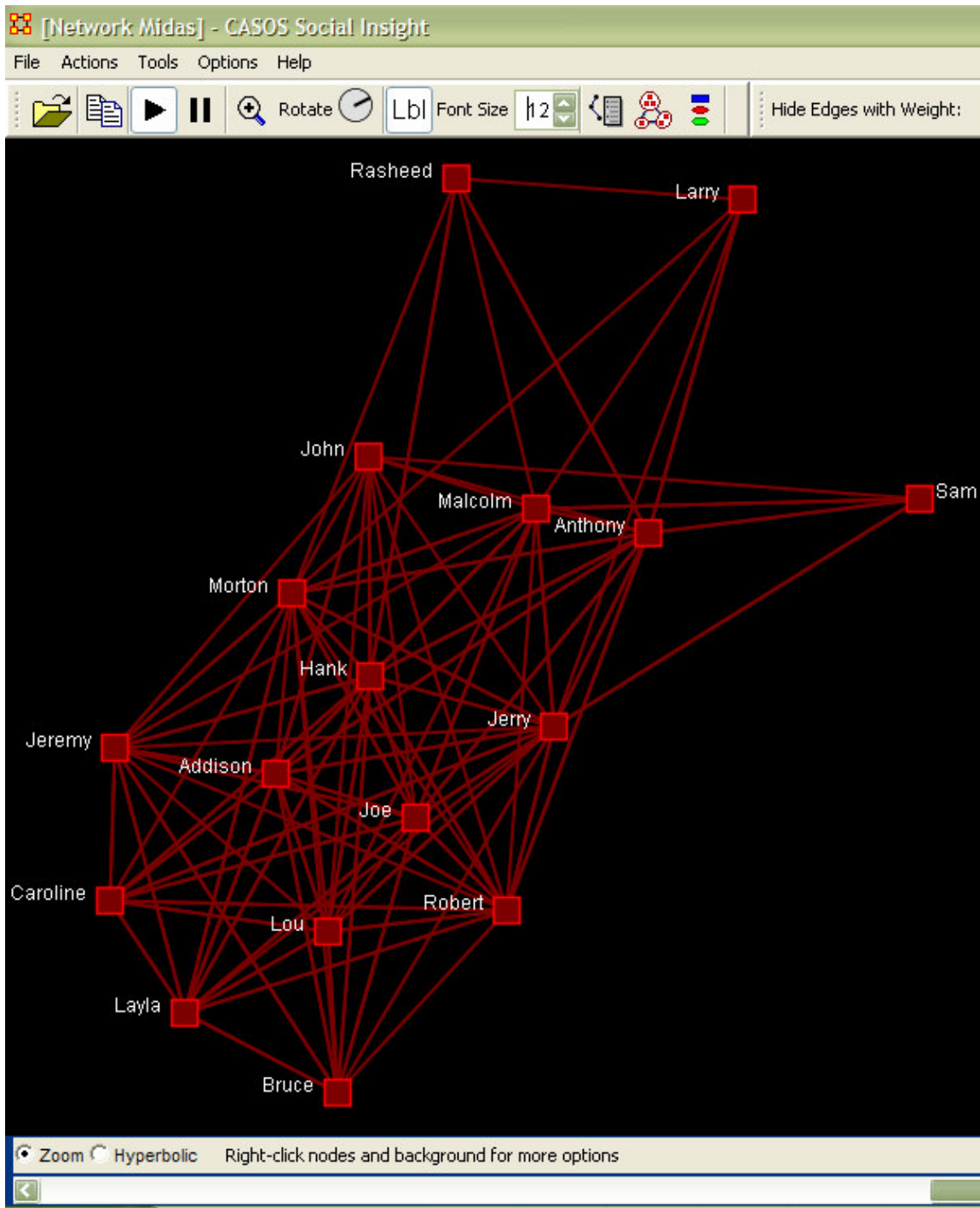


Click "Yes"

Congratulations! You have now created your own MetaMatrix from scratch. Now, simply load your saved CSV file as you would any DynetML file, and work with your data the same way.

[Loading A MetaMatrix](#)

Below is a our MetaMatrix Network Midas rendered in the ORA Visualizer



[Back To Tasks](#)

Running An Over-Time Analysis

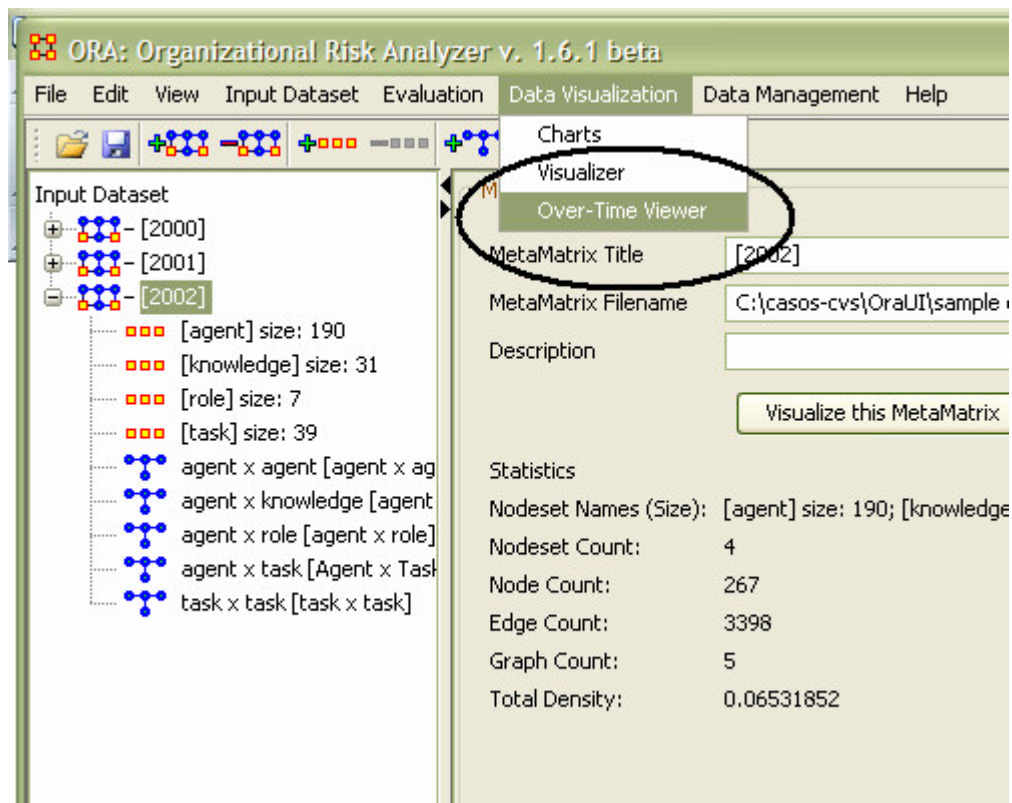
[Overview: Over-Time Viewer](#)

To run the Over-Time Viewer, you must first load MetaMatrices that relate to different time captures of network data. In the example below, you will see in the left window pane under input dataset three MetaMatrices, *2000*, *2001*, and *2002*, have been loaded.

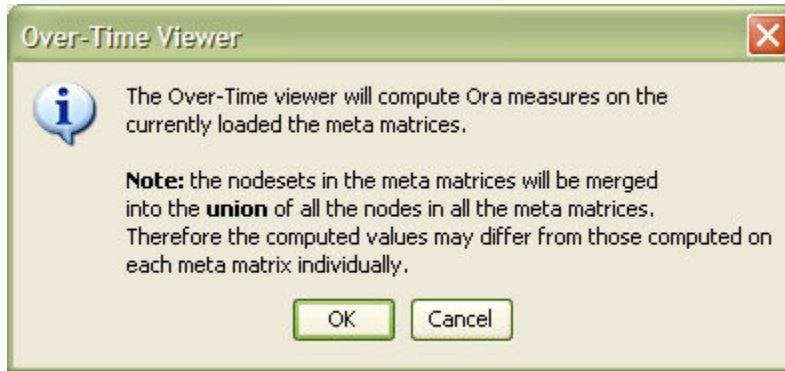
Once you have loaded your MetaMatrices, select the Over-Time Viewer from the main interface tool bar under Data Visualization.

The black ellipse in the screen shot below illustrates how to access the Over-Time Viewer from ORA's main interface.

From the drop-down menu: Data Visualization > Over Time Viewer

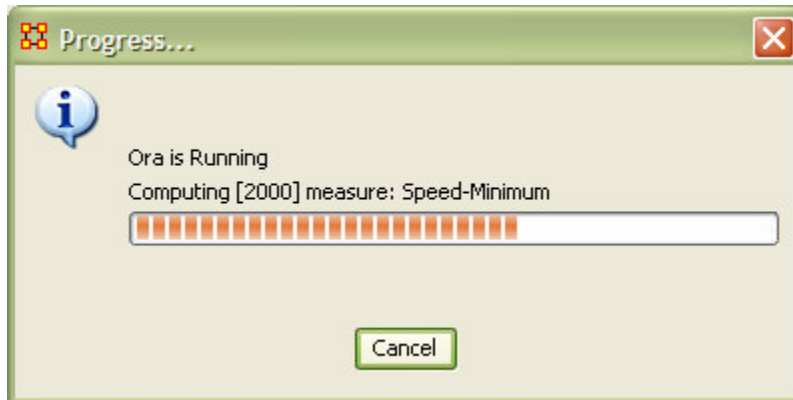


After you select Over-Time Viewer from the drop-down menu, the following information pop-up window will appear:



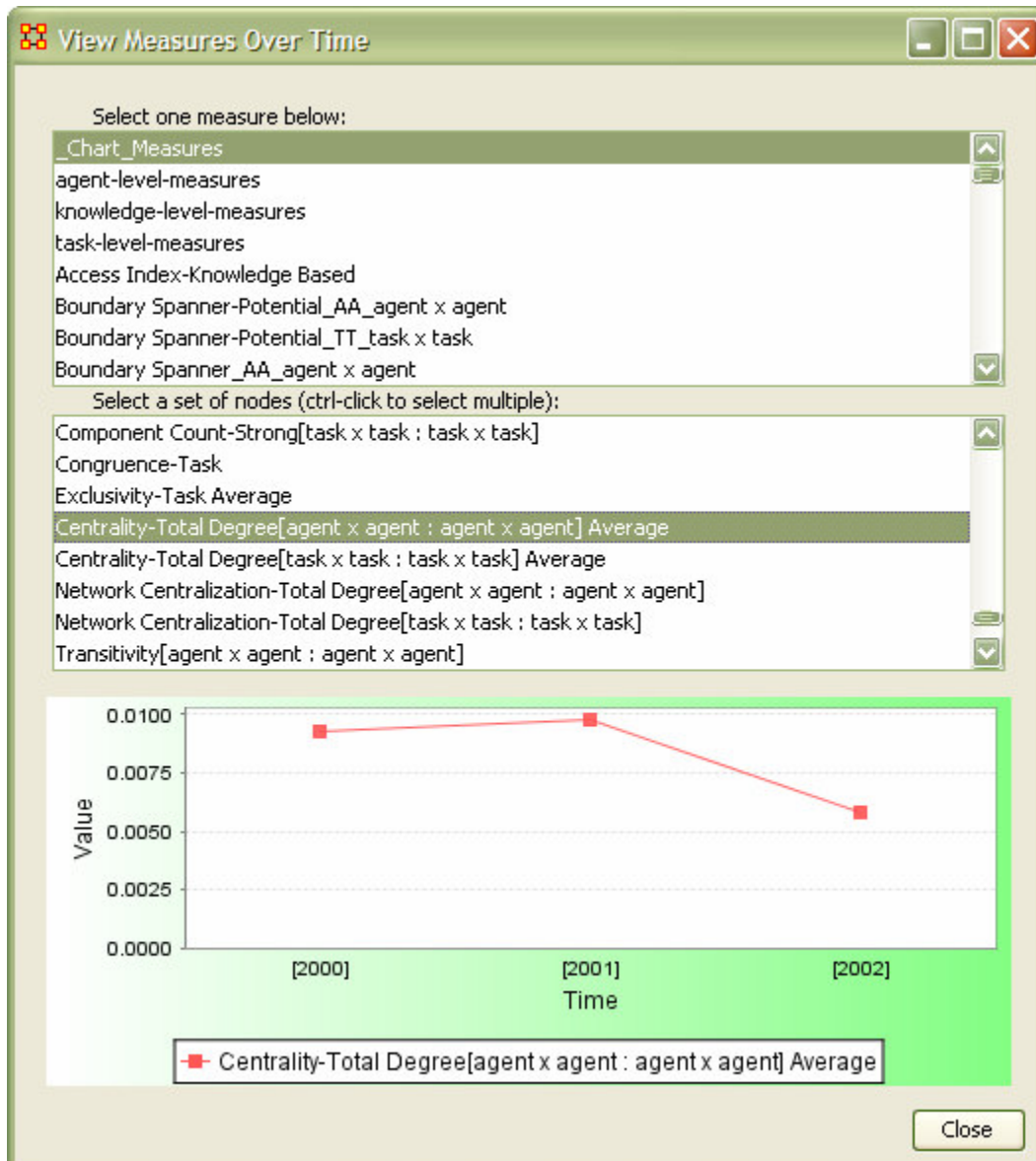
Click **OK**.

The Over-Time Viewer will merge your loaded MetaMatrices. In this case *2000*, *20001*, and *2002* will merge. The *Progress...* pop-up window (below) details the status as ORA runs measures on the merged MetaMatrices:



When it is finished running, the *View Measures Over Time* window appears next (screen shot below).

Here you can run specific measures against the nodesets within your merged MetaMatrices. In the example below, we see that we have opted to run the measure *_Chart_Measures* against the node set *Centrality-Total Degree*. Multiple sets of nodes can be selected against a measure by using "ctrl-click."



The results are displayed in the chart at the bottom of the View Measures Over Time window (shown above).

Based on this analysis, we can conclude the overall Centrality-Total Degree measure of our network sample had increased slightly from 2000 to 2001, then decreased noticeably in 2002.

References

- Ashworth, M. and K. M. Carley, 2003, *Critical Human Capital*, Working Paper, CASOS, Carnegie Mellon, Pittsburgh PA.
- Bonacich, Phil 1987, *Power and centrality: A family of measures*, American Journal of Sociology 92: 1170-1182.
- Borgatti, S.P. 2003, *The Key Player Problem*. Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers, R. Breiger, K. Carley, & P. Pattison (Eds.) Committee on Human Factors, National Research Council, 241-252.
- Burt, Ronald, *Structural Holes: The Social Structures of Competition*. Cambridge, MA: Harvard University Press, 1992.
- Carley, Kathleen 2002, *Summary of Key Network Measures for Characterizing Organizational Architectures*, Unpublished Document: CMU 2002
- Cormen, Leiserson, Rivest, Stein 2001, *Introduction to Algorithms, Second Edition*, Cambridge, MA: MIT Press, 2001.
- Carley, K, Dekker, D., Krackhardt, D (2000). *How Do Social Networks Affect Organizational Knowledge Utilization?*
- Fienberg, S.E., Meyer, M.M., and Wasserman, S.S. (1985), *Statistical Analysis of Multiple Sociometric Relations*, Journal of the American
- Freeman, L.C. (1979), *Centrality in Social Networks I: Conceptual Clarification*. Social Networks, 1, 215-239.
- Krackhardt, D. 1994, *Graph Theoretical Dimensions of Informal Organizations*, In Computational Organization Theory, edited by Carley, Kathleen M. and M.J. Prietula, Hillsdale, NJ: Lawrence Erlbaum Associates, 1994.
- Krackhardt, D. 1998, *Simmelian Tie: Super Strong and Sticky*. In Roderick Kramer and Margaret Neale (eds.) Power and Influence in Organizations. Thousand Oaks, CA: Sage, pp. 21-38, 1998.
- Latora, V., Marchiori, M, *Efficient Behavior of Small-World Networks* PHYS REV LETT 87(19): NOV 5 2001
- Newman MEJ, Moore C., Watts DJ, *Mean-field solution of the small-world network model* , PHYS REV LETT 84 (14): 3201-3204 APR 3 2000
- Newman MEJ, Watts DJ, *Renormalization group analysis of the small-world network model*, PHYS LETT A 263 (4-6): 341-346 DEC 6 1999
- Newman MEJ, Watts DJ, *Scaling and percolation in the small-world network model*, PHYS REV E 60 (6): 7332-7342 Part B DEC 1999 Statistical Association}, 80, 51-67.
- Wasserman, Stanley and Katherine Faust, *Social Network Analysis: Methods and Applications*, Cambridge: Cambridge University Press, 1994.
- Watts DJ, *Networks, dynamics, and the small-world phenomenon*, AM J SOCIOL 105 (2): 493-527 SEP 1999
- Watts DJ, Strogatz SH, *Collective dynamics of 'small-world' networks*, NATURE 393 (6684): 440-442 JUN 4 1998

Additional Resources for CASOS tools and this tool chain:

- Kathleen M. Carley, Jana Diesner, Jeffrey Reminga, Maksim Tsvetovat, 2004, *An Integrated Approach to the Collection and Analysis of Network Data*, In Proceedings of the NAACSOS 2004 Conference, Pittsburgh, PA
- Kathleen M. Carley, 2004, *Estimating Vulnerabilities in Large Covert Networks Using Multi-Level Data*, In Proceedings of the NAACSOS 2004 Conference, Pittsburgh, PA
- Kathleen M. Carley, 2003, *Dynamic Network Analysis in Dynamic Social Network Modeling and Analysis: Workshop Summary and Papers*.
- Ronald Breiger, Kathleen Carley, and Philippa Pattison, (Eds.) *Committee on Human Factors*, National Research Council, National Research Council. Pp. 133-145, Washington, DC.
- Kathleen M. Carley, Jana Diesner, Jeffrey Reminga, Maksim Tsvetovat, 2005-forthcoming, *Toward an Interoperable Dynamic Network Analysis Toolkit*, *DSS Special Issue on Cyberinfrastructure for Homeland Security: Advances in Information Sharing, Data Mining, and Collaboration Systems*.

Where to find out more on SNA

- Scott, John, 2000, *Social Networks*, Sage (2nd edition)
- Wasserman, S. & K. Faust, 1994, *Social Network Analysis: Methods and Applications*

On the Web

CASOS: Center for Computational Analysis of Social and Organizational Systems (<http://www.casos.cs.cmu.edu/index.html>)