

User Manual and Installation Guide for the Graph Matching Toolkit (GMT) Version 1.0

by Heather Roy, Kirk Ogaard, and Sue Kase

ARL-TR-6772

January 2014

NOTICES

Disclaimers

The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturer's or trade names does not constitute an official endorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.

Army Research Laboratory

Aberdeen Proving Ground, MD 21005-5067

ARL-TR-6772

January 2014

User Manual and Installation Guide for the Graph Matching Toolkit (GMT) Version 1.0

**Heather Roy, Kirk Ogaard, and Sue Kase
Computational and Information Sciences Directorate, ARL**

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188		
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing the burden, to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) January 2014		2. REPORT TYPE Final		3. DATES COVERED (From - To) July 2012–May 2013	
4. TITLE AND SUBTITLE User Manual and Installation Guide for the Graph Matching Toolkit (GMT) Version 1.0			5a. CONTRACT NUMBER		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S) Heather Roy, Kirk Ogaard, and Sue Kase			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) U.S. Army Research Laboratory ATTN: RDRL-CII-C Aberdeen Proving Ground, MD 21005-5067			8. PERFORMING ORGANIZATION REPORT NUMBER ARL-TR-6772		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT The intelligence community today is combating information overload. Analysts lack tools to extract relevant information from large masses of information. Graph-based algorithms can assist analysts to sift through vast amounts of information in order to find the subset relevant to a common intelligence picture. Graph-based tools can support social network analysis by facilitating the reasoning over relationships between actors and groups of actors. In order to address the challenge of information overload, researchers at the U.S. Army Research Laboratory developed the Graph Matching Toolkit (GMT). GMT is a visual interface for performing graph matching and serves as a front end to a variety of graph matching algorithms such as the Truncated Search Tree (TruST) algorithm. This report demonstrates the GMT interface and TruST algorithm in the context of performing a series of queries on the Ali Baba Data Set. Descriptions of key features and capabilities, as well as, step-by-step instructions for GMT use are provided. An extended use case suggests GMT is an effective tool for facilitating an analyst's search and social network analysis of a complex dataset to rapidly identify high-value targets (appendix A). An installation guide for GMT (appendix B) and technical support contact information (appendix C) are also included.					
15. SUBJECT TERMS graph matching toolkit, GMT, user guide, social network analysis, TruST algorithm, Ali Baba Data Set					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 48	19a. NAME OF RESPONSIBLE PERSON Sue Kase
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (Include area code) 410-278-3011

Contents

Contents	iii
List of Figures	v
List of Tables	vii
Acknowledgments	viii
1. Introduction	1
2. GMT Overview	2
2.1 TruST Algorithm.....	3
2.2 User Interface	4
2.3 Ali Baba Data Set	5
3. Data Sources	5
3.1 Selecting a Data Source	5
3.2 Installing a New Data Source.....	6
4. Creating a Search Network	8
4.1 Adding Nodes.....	8
4.2 Adding Links.....	9
4.3 Assigning Node Categories	10
4.4 Adding Text to Nodes.....	11
4.5 Assigning Link Categories.....	12
4.6 Types of Link Categories.....	12
5. Selecting a Search Algorithm	14
6. Understanding GMT Search Results	14
6.1 Setting User Preferences for Search Results	15
7. Increasing Network Search Size	16

7.1 Three-Node Network Search Results	17
8. Saving a Search	19
9. Starting a New Search	20
10. Discussion	21
11. Conclusions and Future Revisions	22
12. References	23
Appendix A. Graph Matching Toolkit Use Case: Extended Ali Baba Example Query	25
Appendix B. Graph Matching Tool Installation Guide	31
Appendix C. Graph Matching Tool Support Contact Information	37
Distribution List	39

List of Figures

Figure 1. Flowchart of GMT and TruST algorithm components.....	3
Figure 2. GMT main window.	4
Figure 3. Selecting a Data Source.	5
Figure 4. Installing a new Data Source.....	6
Figure 5. Selecting the XML file for the dataset.....	6
Figure 6. Selecting the Node Ontology for the dataset.....	7
Figure 7. Selecting the Link Ontology for the dataset.....	7
Figure 8. Setting the Name for the dataset.....	8
Figure 9. Creating and adding nodes.....	8
Figure 10. Drawing nodes on the GMT canvas.	9
Figure 11. Adding links.....	9
Figure 12. Adding links and link direction.	10
Figure 13. Setting node categories.	10
Figure 14. Adding text to a node.....	11
Figure 15. Using the text box to enter text for a node.	11
Figure 16. Selecting link categories.	12
Figure 17. Link legend.	13
Figure 18. Color coded link (e.g., purple = type).....	13
Figure 19. Selecting a search algorithm.	14
Figure 20. Search results.....	15
Figure 21. Search result Preferences.	16
Figure 22. Three-node search network and results.....	17
Figure 23. Zoomed-in view of results.	18
Figure 24. Corresponding messages from the ARL Ali Baba Data Set.	19
Figure 25. Saving searches.	20
Figure 26. Opening saved searches.	20
Figure 27. Starting a New Search.....	21
Figure A-1. Police Report 60.	26
Figure A-2. Two-node template graph for a search (left side) and search results (right side).	27
Figure A-3. The messages from the dataset corresponding to the search results.....	28

Figure A-4. AXIS Pro concept map of GMT search results outlined by TacRep number.	29
Figure A-5. GMT three-node search template.	29
Figure A-6. GMT fuzzy match result: TacRep 50	30
Figure A-7. AXIS Pro link diagram of the power hierarchy of the Ali Baba Group.	30
Figure B-1. The first step for the GMT installation wizard.	33
Figure B-2. The second step for the GMT installation wizard.	34
Figure B-3. The third step for the installation wizard.	34
Figure B-4. The fourth step for the installation wizard.	35

List of Tables

Table B-1. The description of the fields found in the GMT configuration file.	33
--	----

Acknowledgments

This research was supported in part by an appointment to the Student Research Participation Program at the U.S. Army Research Laboratory (ARL), administered by the Oak Ridge Institute for Science and Education through an interagency agreement between the U.S. Department of Energy and ARL.

1. Introduction

The problem in the intelligence community today is often not a lack of information but rather information overload. Analysts lack tools to locate the relatively few bits of relevant information in ever-increasing data sets and tools to support reasoning over that information. Graph-based algorithms can help analysts sift through a large amount of data to find the small subset that is potentially relevant to a common intelligence picture. Graph-based tools can facilitate reasoning over relationships between actors and groups of actors in contrast to spreadsheets and file-based databases that facilitate reasoning at an individual actor level. Enabling analysis at the relationship level increases the probability of detecting suspicious behavior such as the planning stages of an improvised explosive device attack when early warning signs are not obvious but take on the appearance of activities and behavior typically found in the context of everyday life.

Graph-based algorithms allow large graph databases (called data graphs) to be searched for instances of a specific pattern graph called a subgraph. For example, an analyst defines a subgraph pattern that is believed to be indicative of an activity of interest. This subgraph pattern is structured as a classical graph format with objects (also called nodes or entities) interrelated by relationships (also called edges or links). Nodes can represent people, organizations, locations, individuals, or facilities. Links represent relationships such as communication, participation, or association. In addition to nodes and links, attributes (or labels) can store details about objects and their relationships. Once the subgraph pattern is defined, a graph matching algorithm is executed to search the data graph and identify pattern matches. Without this type of graph-based search approach, an analyst's task of identifying an activity of interest within a large body of evidence is much more difficult and time-consuming.

Because analysts operate in an environment with limited observability, sometimes searching for an exact one-to-one correspondence between a subgraph pattern and a data graph is too difficult. In addition, the analyst might need to match a general pattern without knowing all of the details, or the analyst may have defined some aspects of the pattern incorrectly. Therefore, being able to find inexact pattern matches is critical. This necessitates inexact (or fuzzy) graph matching where two graphs are compared even though they are semantically or topologically different. Fuzzy graph matching between the subgraph pattern and the data graph allows the analyst to focus on a set of most likely situations.

This report introduces the Graph Matching Toolkit (GMT), a visual interface for performing graph matching. GMT serves as a front end to any variety of graph matching algorithms. As a first test case, the Truncated Search Tree (TruST) algorithm (1) was used to evaluate the interface. TruST is a heuristic graph matching algorithm developed at the State University of

New York at Buffalo Center for Multisource Information Fusion. The GMT interface provides an easy to use link between a human searching for meaningful patterns in very large and complex data.

This report is organized as follows. We first introduce the GMT and the linking mechanism with the TruST algorithm. Then we describe the GMT visual user interface and supported features. Next we provide an overview of the GMT programming platform. In section 2.3, we introduce the context for the examples used throughout the user guide, the Ali Baba Data Set. Section 3 begins the user guide by demonstrating how to select a data source to search. Section 4 illustrates how to construct a simple two-node subgraph pattern. Executing a search using the TruST algorithm is explained in section 5, and an explanation of the search results is provided in section 6. Options associated with search size, saving, and starting a new search are explained in sections 7, 8 and 9, respectively. We conclude with discussion and conclusions in sections 10 and 11, respectively.

In addition, several appendixes are offered in conjunction with this report. In appendix A, a detailed use case is provided that applies GMT's multinodal search capabilities to an intelligence analysis context for the identification of a terrorist social network. An installation guide for GMT is provided in appendix B, with technical support contact information listed in appendix C.

2. GMT Overview

The GMT is implemented in the C++ language using the Qt cross-platform application framework (2) for its graphical user interface. The front end for GMT has been successfully tested on multiple operating system platforms. GMT interfaces with the graph matching algorithm via flat files and stores the subgraph query in a GraphML file prior to executing the TruST graph matching algorithm. The TruST algorithm loads the GraphML file created by GMT, executes the specified subgraph query, and stores the results in an extensible markup language (XML) file. As a last step, GMT loads the XML file containing the results from the TruST algorithm and displays the top k -matching subgraphs with the highest overall scores in the results window. The flowchart in figure 1 illustrates the two main components involved in GMT's graph matching process: (1) the front end enables the analyst to construct the subgraph query, and (2) the back end executes the specified subgraph query using the selected graph matching algorithm.

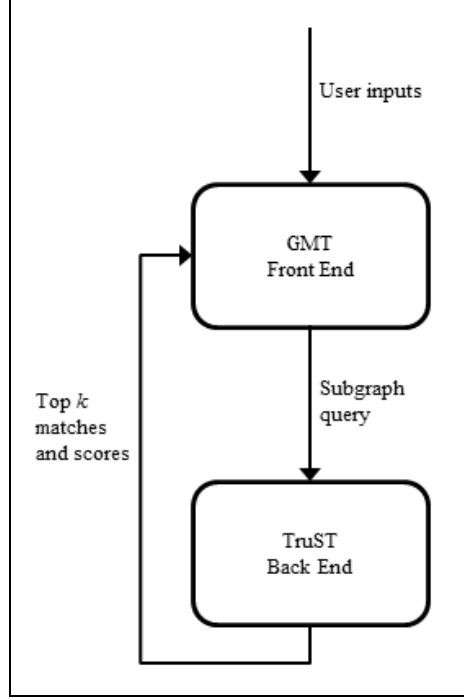


Figure 1. Flowchart of GMT and TruST algorithm components.

2.1 TruST Algorithm

The second component in figure 1 is the TruST back end. A subgraph pattern is constructed using the GMT user interface and then the TruST algorithm performs the subgraph matching. TruST (3, 4) is a heuristic search algorithm optimization of best-first search. The search tree is developed dynamically during the search and initially consists of only the root. At each iteration of the algorithm, a subproblem is selected for exploration from the pool of live subproblems using scores of the current match. TruST uses a strategy similar to the breadth first search strategy found in the literature (5). The basic principle is to process all the nodes at one level of the search tree before processing any node at a deeper level.

The TruST graph matching heuristic (1) is able to rank subgraph-data graph matches. In the case of crisp scoring, this ranking is performed by a simple numerical ordering. To properly exploit the fuzzy scores of inexact graph matching, a method of ranking fuzzy numbers must be used. The method used to perform this is the Chen and Chen (6) method for ranking fuzzy numbers with different spreads and performing fuzzy risk analysis.

Graph matching algorithms such as TruST are designed to search data graphs for instances of matching subgraphs. An example of a large data graph is U.S. Army Research Laboratory (ARL) Ali Baba Data Set containing over 2,000 nodes (7).

2.2 User Interface

The first component in figure 1 is the GMT front end or user interface. The user interface allows a user to draw a graph pattern by using a mouse to select nodes, links, and labels from drop-down menus. This intuitive and visual style of graph building enables analysts to quickly and efficiently build a search query using limited information. Search results direct analysts to potentially relevant information, thus aiding the analyst with searching through large quantities of data to extract useful intelligence information.

The GMT user interface or main window is shown in figure 2. The main window contains several subwindows and tools: a menu bar located at the top (i.e., File, Edit, Data, Search, and Help); a canvas located in the center; four drawing tool icons located along the upper left side (i.e., Add Nodes, Add Links, Remove Nodes, and Undo); a link ontology legend in the bottom left corner; and a results subwindow located at the bottom right corner of the main window. The functionality of the subwindows and tools is explained in more detail over the next several sections.

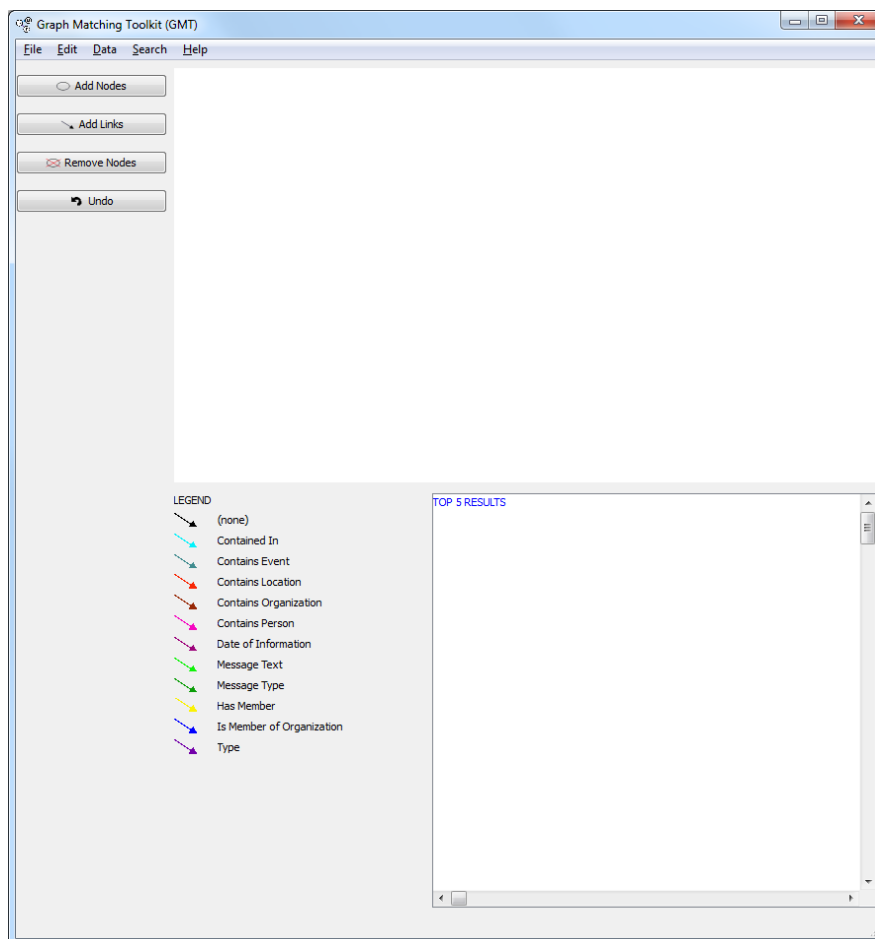


Figure 2. GMT main window.

2.3 Ali Baba Data Set

To demonstrate the capabilities of GMT, an example data graph was constructed of the Ali Baba Data Set. The Ali Baba Data Set was originally developed by Mark Jaworoski and Steve Pavlak for the National Security Agency in 2003 as a series of synthetic databases designed to mimic counterterrorism data (7). These data contain simulated intelligence text reports describing a fictitious terrorist plot in the United Kingdom. The original data set contained 752 simulated text communications documenting the actions of a fictitious suspected terrorist network operating in England intent on bombing a water treatment facility. In 2012, the dataset was revised by members of ARL's Computational and Information Fusion Branch's Social Network Analysis Team (8).

Examples from the ARL Ali Baba Data Set are interwoven throughout this user guide to illustrate the GMT search process. The next section begins the user guide describing how to construct an example search for a person in the Ali Baba Data Set called Yakib Abbaz.

3. Data Sources

3.1 Selecting a Data Source

Before creating a subgraph query, a search must be initiated by selecting the Data option from the menu bar. From the Data drop-down menu, the Data Source option is then selected. This option will display the data sources connected to GMT that are available for searching. In this example, the analyst selects the Ali Baba (ARL) Dataset from the list (figure 3). This is the ground truth version of the Ali Baba Data Set that was created manually at ARL. The other versions of the Ali Baba Data Set available in GMT were automatically generated by Natural Language Processing (NLP) tools.

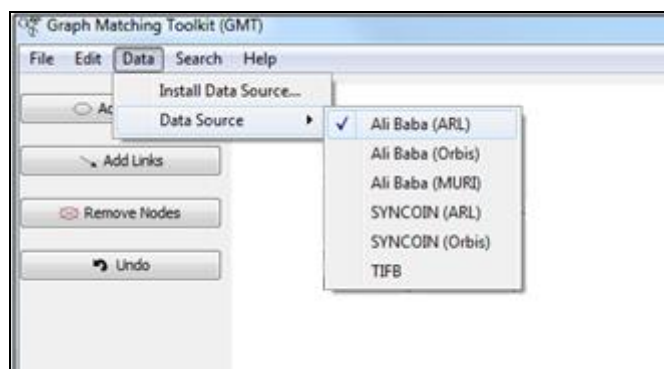


Figure 3. Selecting a Data Source.

3.2 Installing a New Data Source

Additional datasets may be installed using the GMT menu. To install a new dataset, select the Data option from the menu located at the top of the screen. From the Data drop-down menu, select Install Data Source..., shown in figure 4. This option will display a series of three file selection boxes and a text box for configuring the installation of the new dataset. For the first file selection box, choose the XML data file for the dataset (figure 5). For the second file selection box, choose the text file that specifies the node ontology for the dataset (figure 6). For the third file selection box, choose the text file that specifies the link ontology for the dataset (figure 7). In the final text box, specify the name of the dataset (figure 8). Once the configuration of the new dataset is completed, the necessary files will be installed in the GMT installation directory.

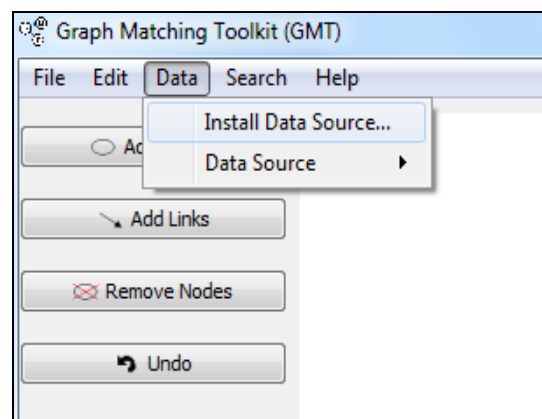


Figure 4. Installing a new Data Source.

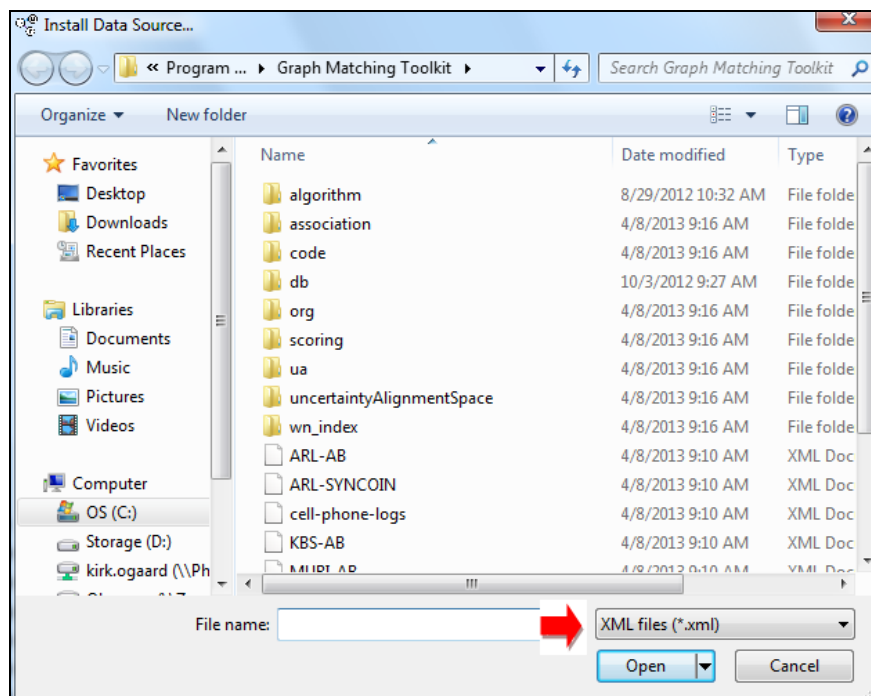


Figure 5. Selecting the XML file for the dataset.

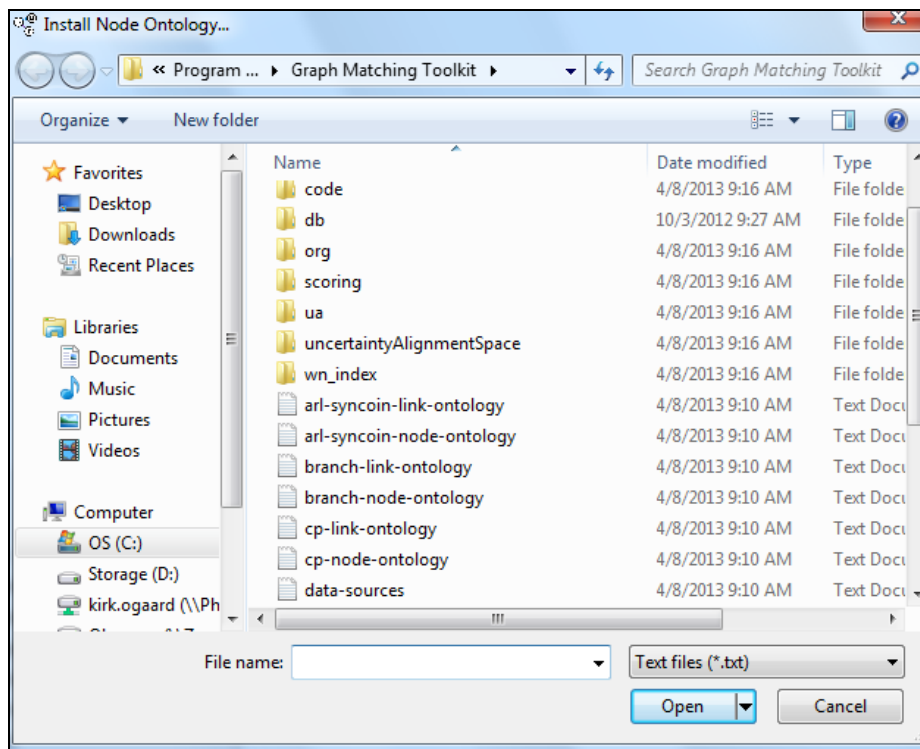


Figure 6. Selecting the node ontology for the dataset.

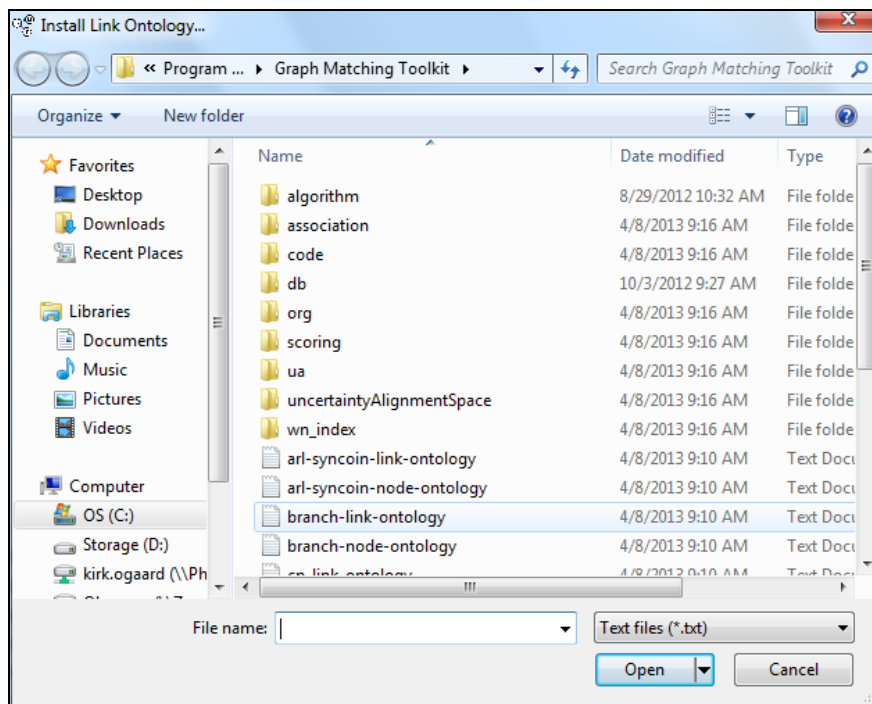


Figure 7. Selecting the link ontology for the dataset.

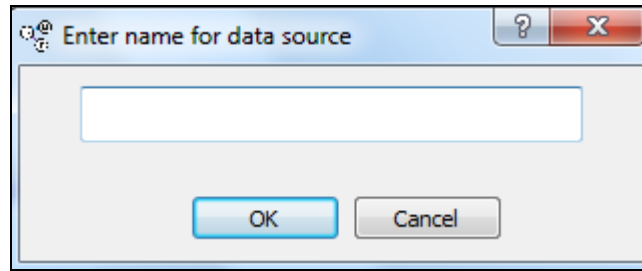


Figure 8. Setting the name for the dataset.

4. Creating a Search Network

4.1 Adding Nodes

To demonstrate the search capabilities offered by GMT, we simulate the context of an analyst using GMT to determine if a person with the Ali Baba Data Set named Yakib Abbas is a terrorist. The analyst begins by using GMT to create a two-node subgraph to search the Ali Baba data graph for confirming or disconfirming evidence. Nodes are used to represent entities such as persons, locations, organizations, events, messages, or times. There are three methods for creating nodes in GMT. One method is to select Edit on the menu bar and then select Add Nodes to create a new node on the drawing canvas (figure 9A). A second method is to create new nodes by using the shortcut Ctrl+Shift+O located alongside Add Nodes. The third method is to click the Add Nodes icon located in the upper left corner of the main window (figure 9B). Once this icon is activated, nodes can be created by clicking on the canvas at the desired placement point for the node. Multiple nodes may be added at one time when the Add Nodes icon is active (figure 10). Node placement may be adjusted by clicking and dragging the node on the canvas. Any extraneous nodes that are created may be removed by clicking the Remove Nodes icon (figure 10).

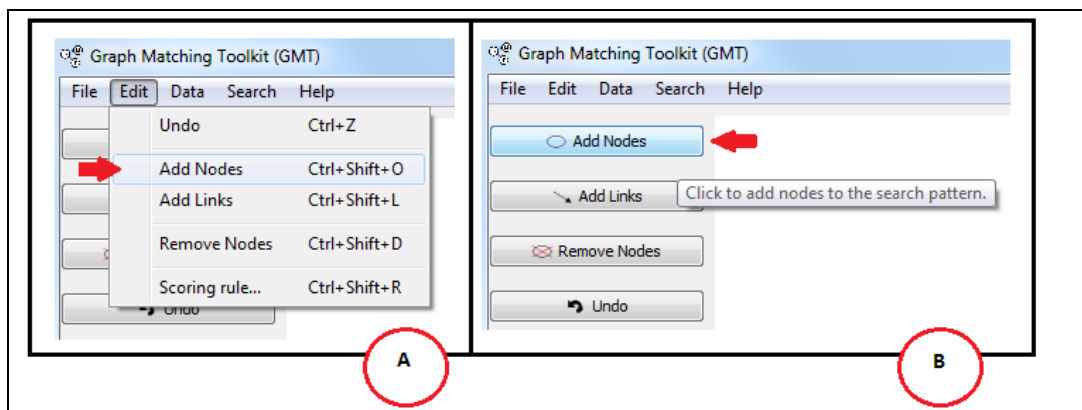


Figure 9. Creating and adding nodes.

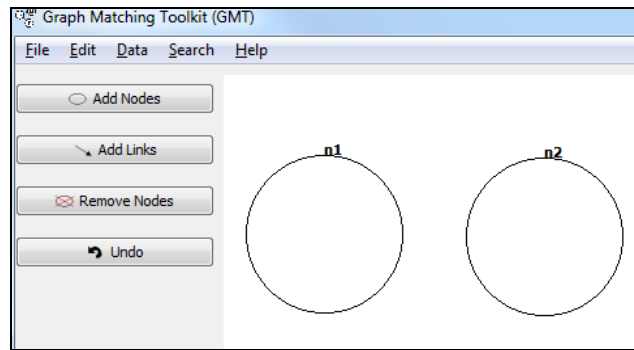


Figure 10. Drawing nodes on the GMT canvas.

4.2 Adding Links

Similar to adding nodes, there are multiple methods for creating links. Links can be added in three ways: selecting the Edit option from the menu bar and then selecting Add Links (figure 11A); using the short cut Ctrl+Shift+L; or clicking on the Add Links icon in the upper left corner of the main window below Add Nodes (figure 11B). Once the Add Links option is activated, links can be added directionally by first clicking in the center of the node from which the link originates and then clicking in the center of the second node to complete the link connection. In figure 12, a link has been drawn that originates from the first node (n1) and connects to the second node (n2). The direction of the node is indicated by the direction of the arrow.

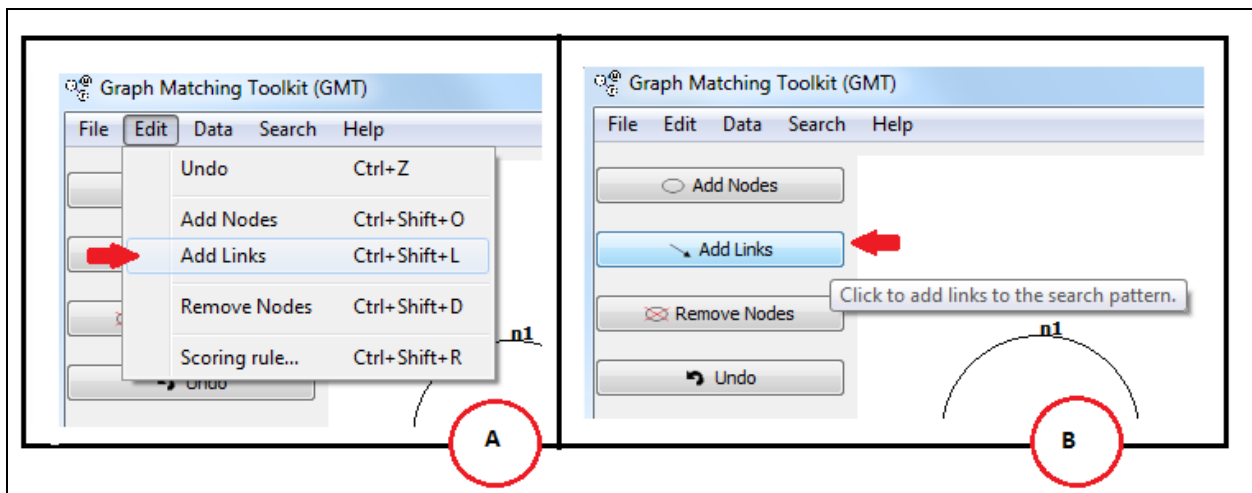


Figure 11. Adding links.

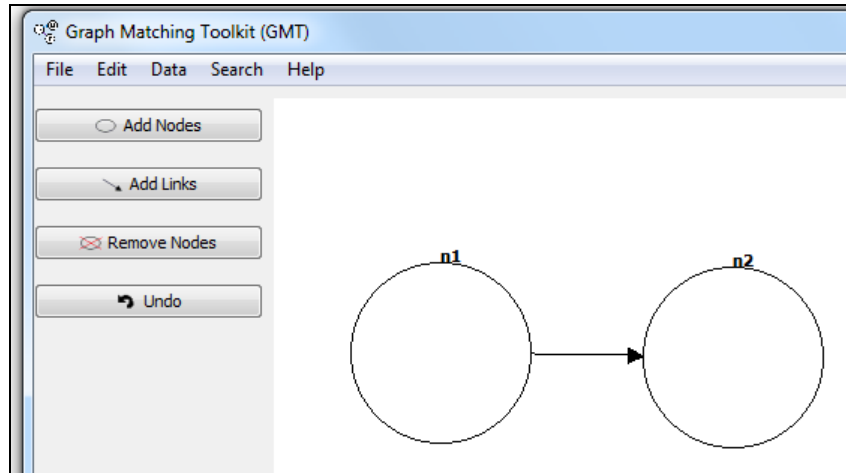


Figure 12. Adding links and link direction.

4.3 Assigning Node Categories

Nodes are usually assigned categories to focus the search. Node categories are selected by first using the mouse to right-click on the node to bring up a drop-down menu. Category options are displayed by selecting the first option from the drop-down menu: set category for node.

In the example involving Yakib Abbas, the analyst selects the Person category (figure 13). A category can be assigned to n2 in a similar manner by right-clicking on the node and following the same drop-down options. The analyst is interested in assessing the validity of Yakib Abbas being a terrorist. Because a terrorist is a type of person, the analyst also assigns n2 the Person category (figure 13).

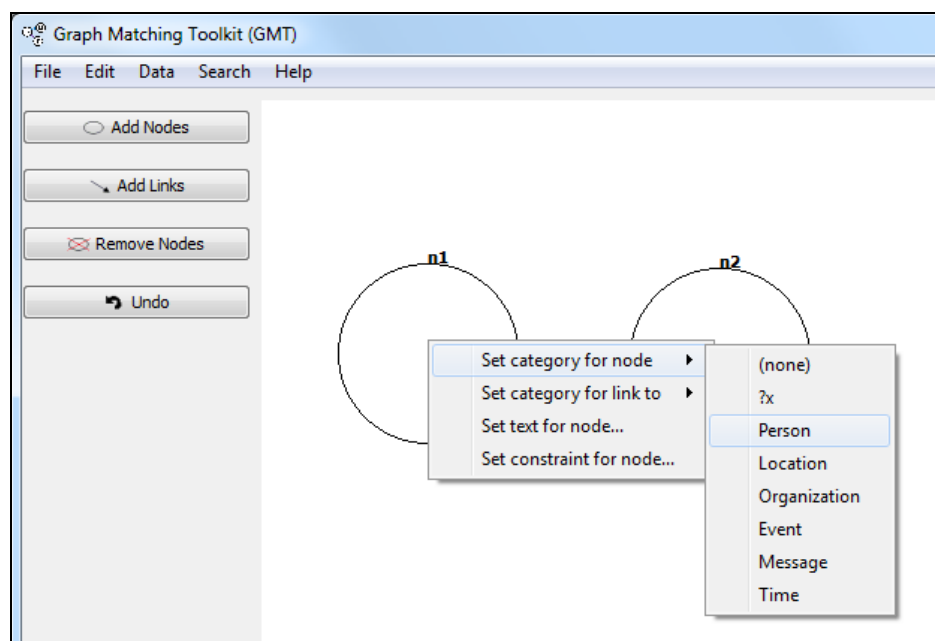


Figure 13. Setting node categories.

4.4 Adding Text to Nodes

In addition to a category, nodes can also be assigned text label by first right-clicking on the node to bring up the drop-down menu and then selecting the third option Set text for node... (figure 14). Selecting this option will bring up a text box. The keyboard is used to type in the text for the node.

In the example, the analyst selected the Person category for n1. In figure 15, the analyst typed in the name Yakib Abbaz in the text box to add a label to n1. The analyst wants to evaluate if Yakib Abbaz is a terrorist; therefore, by following the same steps for adding a name to n1, the analyst assigns n2 the label Terrorist.

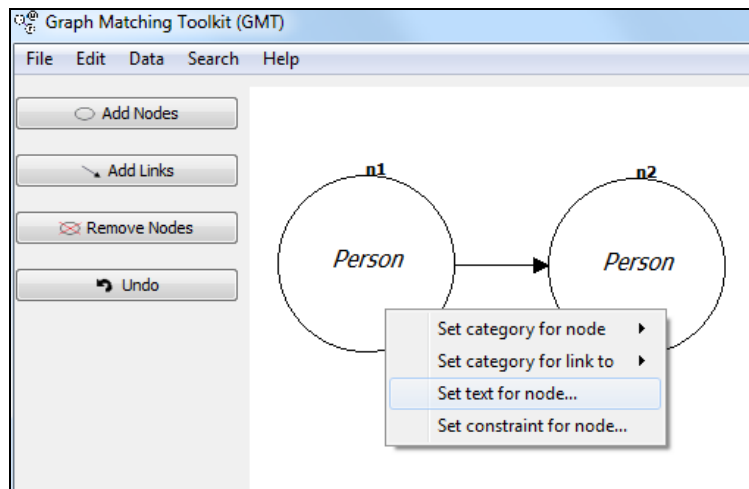


Figure 14. Adding text to a node.

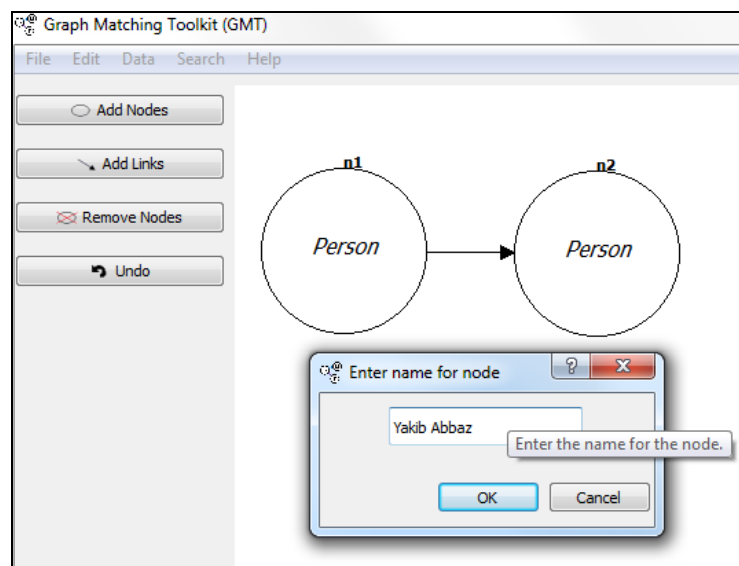


Figure 15. Using the text box to enter text for a node.

4.5 Assigning Link Categories

Similar to nodes, categories can be assigned to links connecting the nodes. Link categories are selected by right-clicking on the node connected to the end of the arrow to display the drop-down menu. From the drop-down menu, the second option, Set category for link to, is selected (figure 16). By selecting this option, the menu will display the node(s) that are connected to the first node (n1). In figure 16, the analyst is setting the category of the link from n1 to n2. When the desired node (ex., n2) is selected, a third drop-down menu providing link categories (figure 16) is displayed.

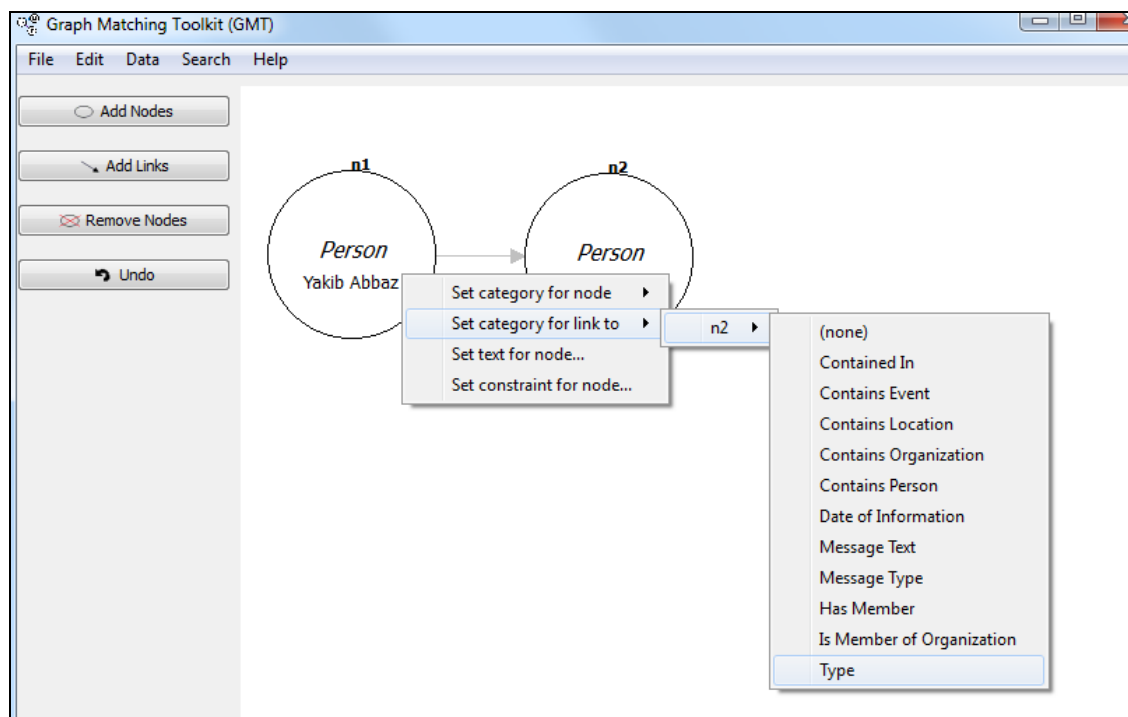


Figure 16. Selecting link categories.

The majority of link categories are message oriented (e.g., Contained In, Contains Event, Contains Location). However, since the analyst is searching for the Person Yakib Abbaz connected to a type of person (Terrorist) the link category Type is selected (figure 16).

4.6 Types of Link Categories

Depending on the data graph, there can be many link categories to choose from. A legend in the bottom left corner of the main window displays link colors and their associated meaning (figure 17). Each link category is assigned a different color; for example, the link category Type is purple. When the analyst assigns the link category Type, the link color changes from black (figure 18A) to purple (figure 18B). Using colors rather than labels to signify the link category reduces the text clutter of the subgraph being constructed.

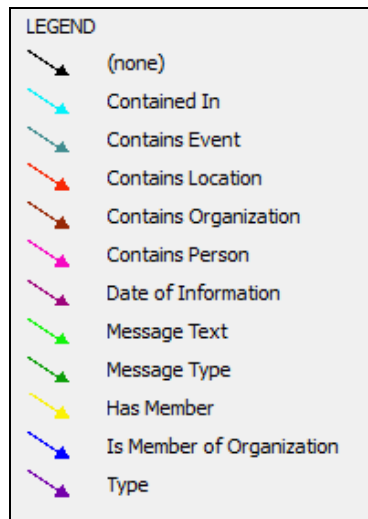


Figure 17. Link legend.

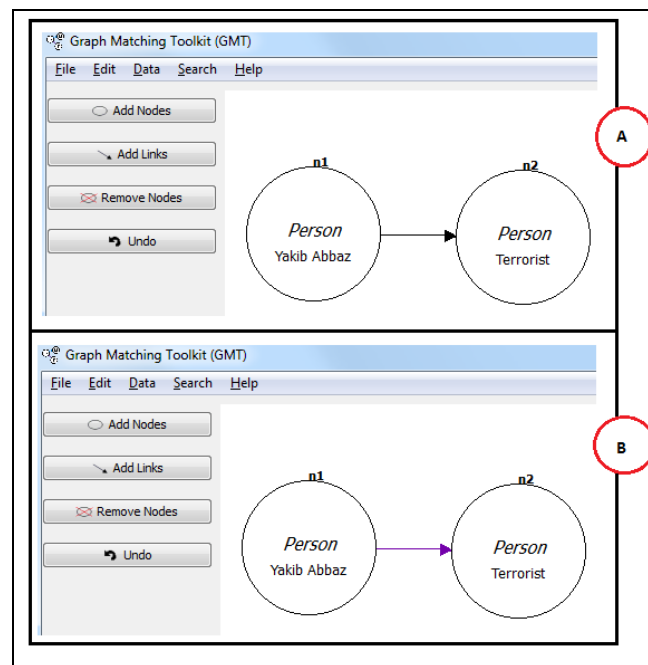


Figure 18. Color coded link (e.g., purple = type).

At this point, the analyst has completed drawing the subgraph pattern to be used as a search query by a subgraph matching algorithm. Figure 18B shows this search query consisting of two nodes (n1 and n2) with categories Person, and text “Yakib Abbas” added to n1 and text Terrorist added to n2. The link connecting the nodes is of category Type. The analyst constructed this query to search the data graph for evidence supporting Yakib Abbas is a terrorist. Before executing the search, the analyst must select the data source (data graph) to search and the type

of subgraph matching algorithm to perform the search. The steps for selecting the Ali Baba Data Set as the data graph and TruST as the subgraph matching algorithm are explained in the next section.

5. Selecting a Search Algorithm

After a data source has been selected and the search query created, the Search option from the menu bar is selected. From Search, select the Search With option and then a specific subgraph matching algorithm. For this example in figure 19, the analyst selects the TruST Algorithm.

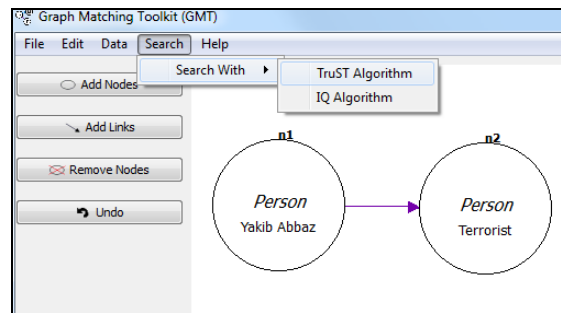


Figure 19. Selecting a search algorithm.

As the search process begins, several command windows (black background with white text) will pop open showing executing script. There may be a short time delay as the algorithm executes the search. The command windows will close when the algorithm has completed the search. The search results will be displayed in the results window as described in the next section.

6. Understanding GMT Search Results

Continuing with the example, the analyst has selected the TruST algorithm to perform the search on the Ali Baba Data Set. After the search executes, GMT will show the top five scored results in the results window in the lower right corner of the main window. The result window is labeled Top 5 Results in blue (figure 20, left). The search results can be viewed by using the scroll bar on the right side of the results window.

The top five results from the search are summarized on the right side of figure 20. Each of the results is a node pair instance of the data graph that matched the subgraph query to some degree of similarity. The score listed below each of the subgraph matches represents this degree of similarity. Results are scored on a scale from 0.0 to 1.0 with a score of 1.0 representing a perfect

match. From the example GMT search results shown in figure 20, the analyst has discovered evidence supporting that Yakib Abbas is a terrorist (see first node pair under Top 5 Results with Score 1.0). Through fuzzy matching, the search results yielded four additional names that are connected to person type Terrorist. An analyst could use these names to create additional searches to investigate the terrorist network further. However, for the purposes of this example, the analyst is interested in finding out more information pertaining to Yakib Abbas.

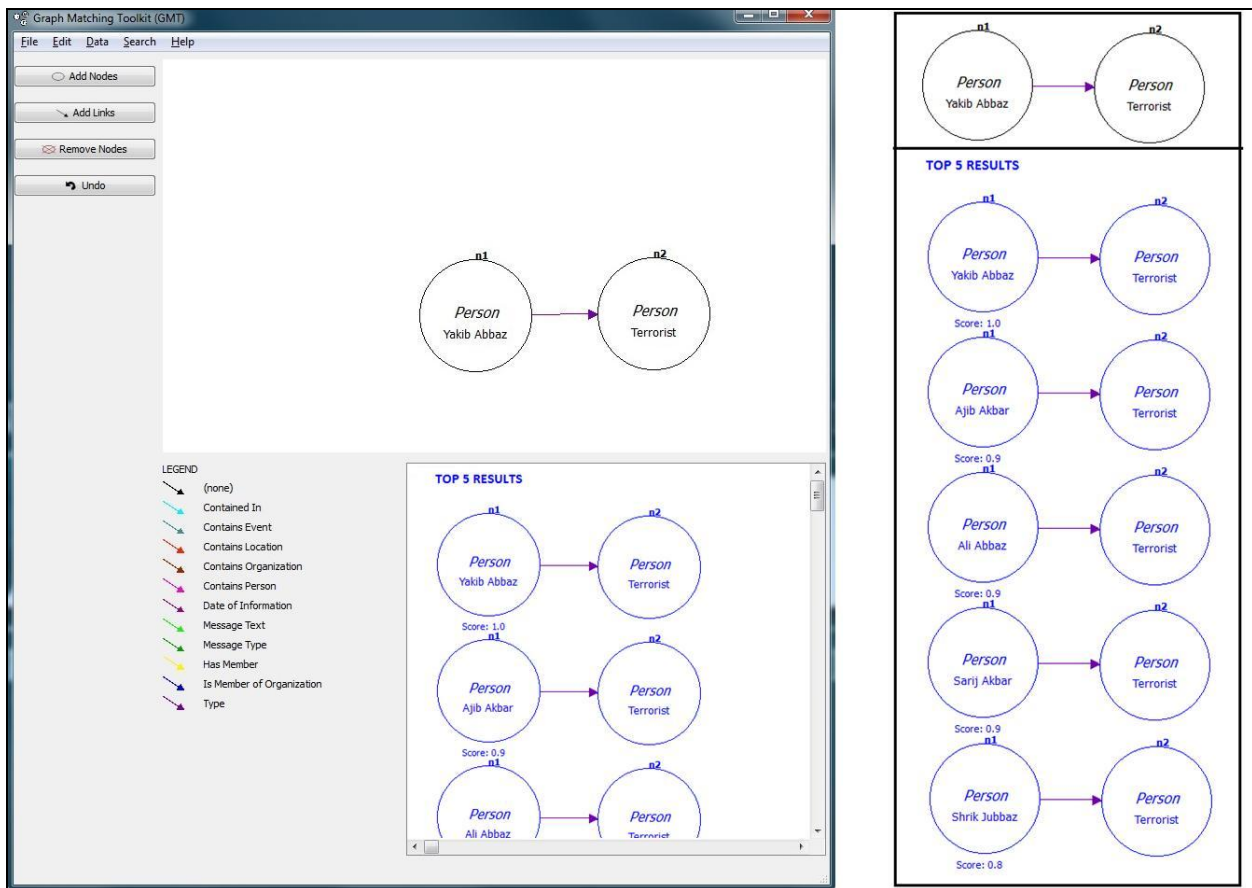


Figure 20. Search results.

6.1 Setting User Preferences for Search Results

Search results are set by default to return the top 5 scored results, but the setting can be modified to allow up to 10 results. The returned results setting is changed by selecting the Preferences... option from under File on the menu bar (figure 21A). When the Preferences... option is selected, a dialog box appears. From the dialog box, the Search tab is selected. The textbox or the arrows can be used to assign how many search results should be returned. Figure 21B shows 10 search results being selected. Once the search result preferences have been set, the results window title Top 5 Results will be updated to display the modified number of results to be displayed. For example, if 10 results are selected to be returned, the title of the results window would appear as Top 5 Results.

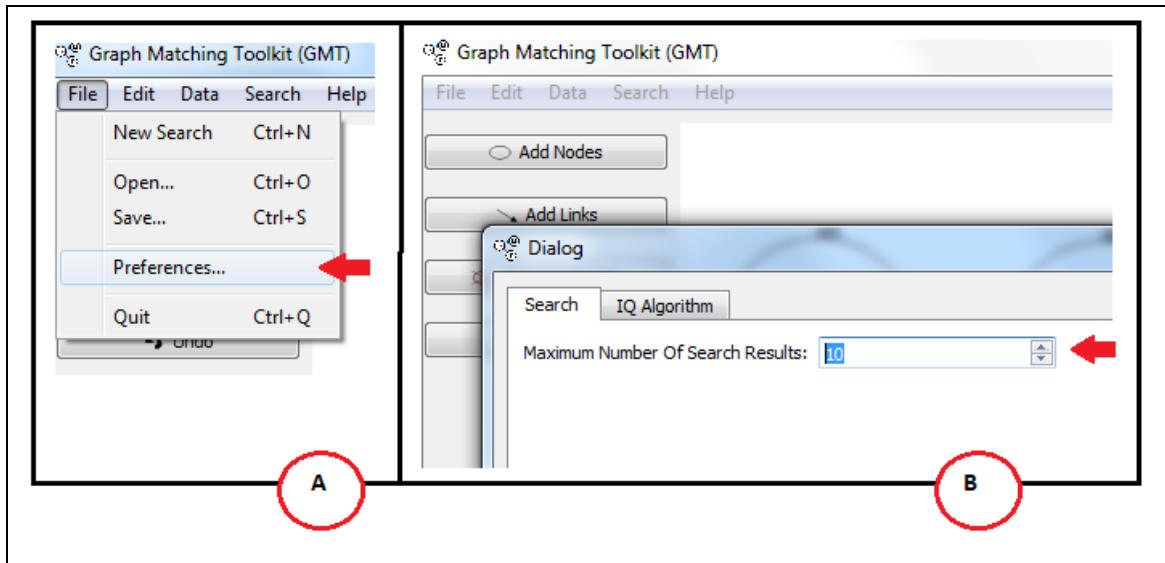


Figure 21. Search result Preferences.

7. Increasing Network Search Size

Using GMT's canvas, search queries can be constructed for a single node, a node pair (previous example), or larger multinode networks. Continuing with the previous example, if the analyst wanted to investigate Yakib Abbaz further, the analyst could create a three-node network to search the data graph. The analyst would start with the original two-node network and then add a third node. In the three-node network, the analyst connects the Yakib Abbaz node (n1) to a third node (n3) categorized as Message. The analyst then categorizes the link connecting n1 to n3 as Contained In. This three-node network will search for Yakib Abbaz, person type Terrorist, contained in messages within the Ali Baba Data Set. The analyst selects the TruST algorithm to perform the search with the top five scored results appearing in the results window. With this larger network query, scrollbars located on the bottom and right side of the results window are used to view all the results (figure 22).

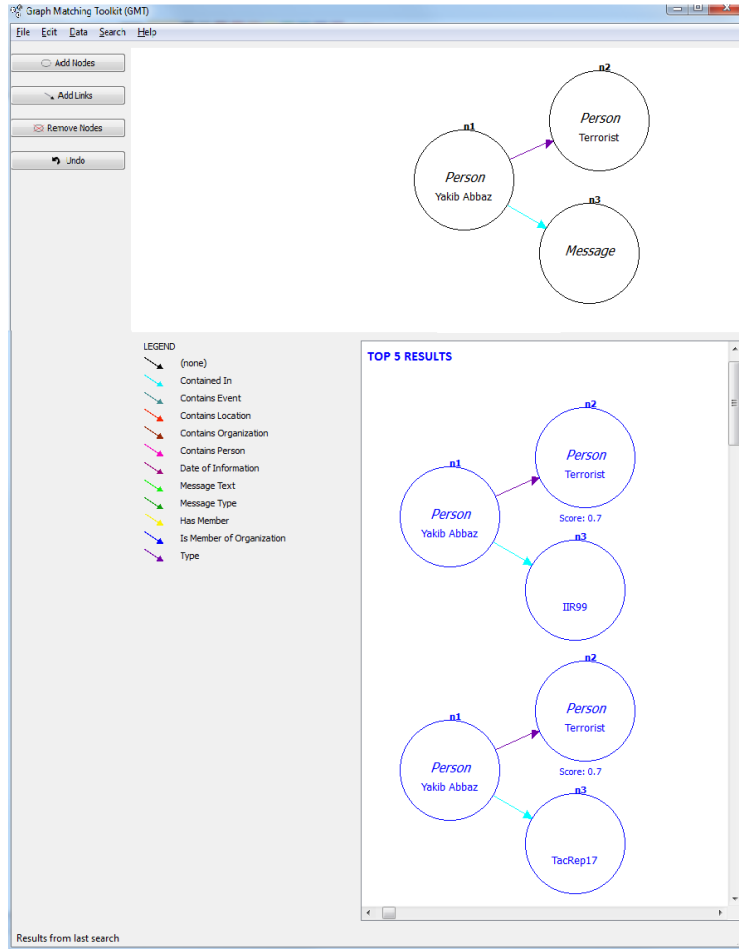


Figure 22. Three-node search network and results.

7.1 Three-Node Network Search Results

The top two scoring results for the three-node search in figure 22 are summarized in figure 23. Using these results, the analyst can quickly locate the important messages in the Ali Baba Data Set to read in detail and evaluate for further action.

For example, the two results displayed in figure 23 show that Yakib Abbas is connected to the messages IIR 99 (Intelligence Information Report 99) and TACREP 17 (Tactical Report 17). Now the analyst can access these specific messages in the Ali Baba Data Set to view the full content of the message text (figures 24A and 24B). From the IIR 99 message, the analyst learns Yakib Abbas is connected to a person named Saleh Sarwuk and that they are involved in “suspicious ‘charitable financing’ activities” (figure 24A). From TACREP 17, the analyst learns Yakib Abbas is connected to a person named Salam, and Salam seems to play some role regarding finances (figure 24B). The analyst also learns that Salam is connected to a person named Abdul and a Sheikh. Additionally, the analyst infers from the messages that Yakib Abbas is operating in London.

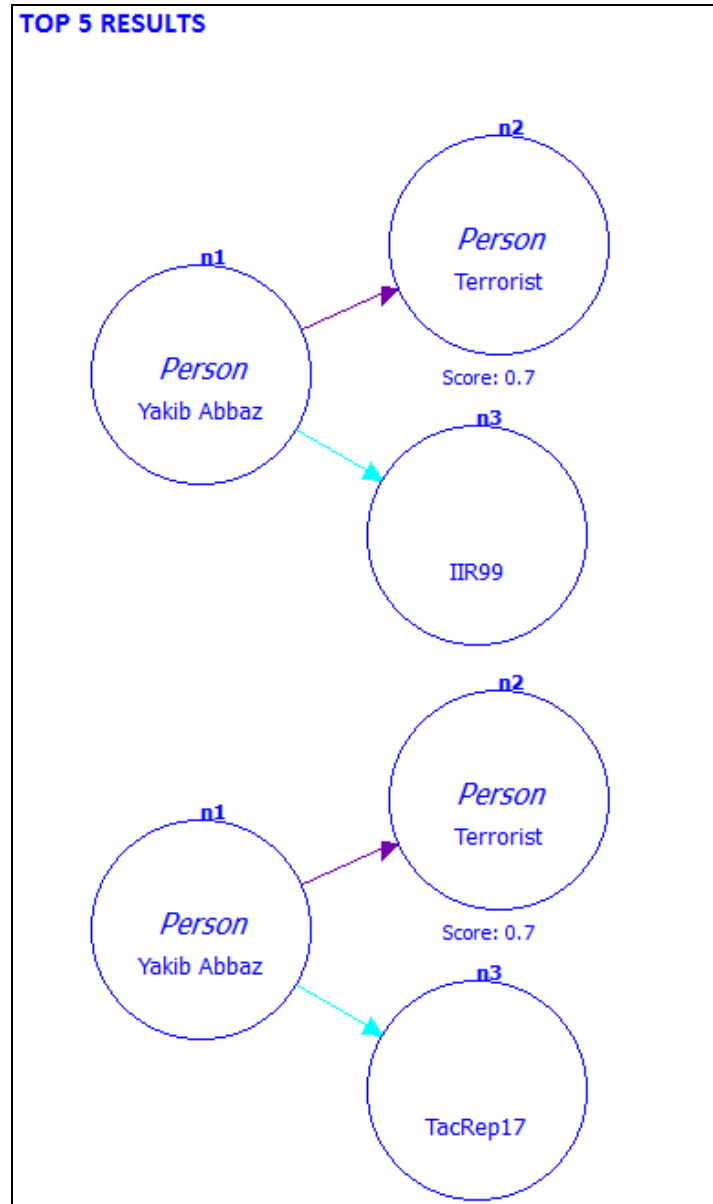


Figure 23. Zoomed-in view of results.

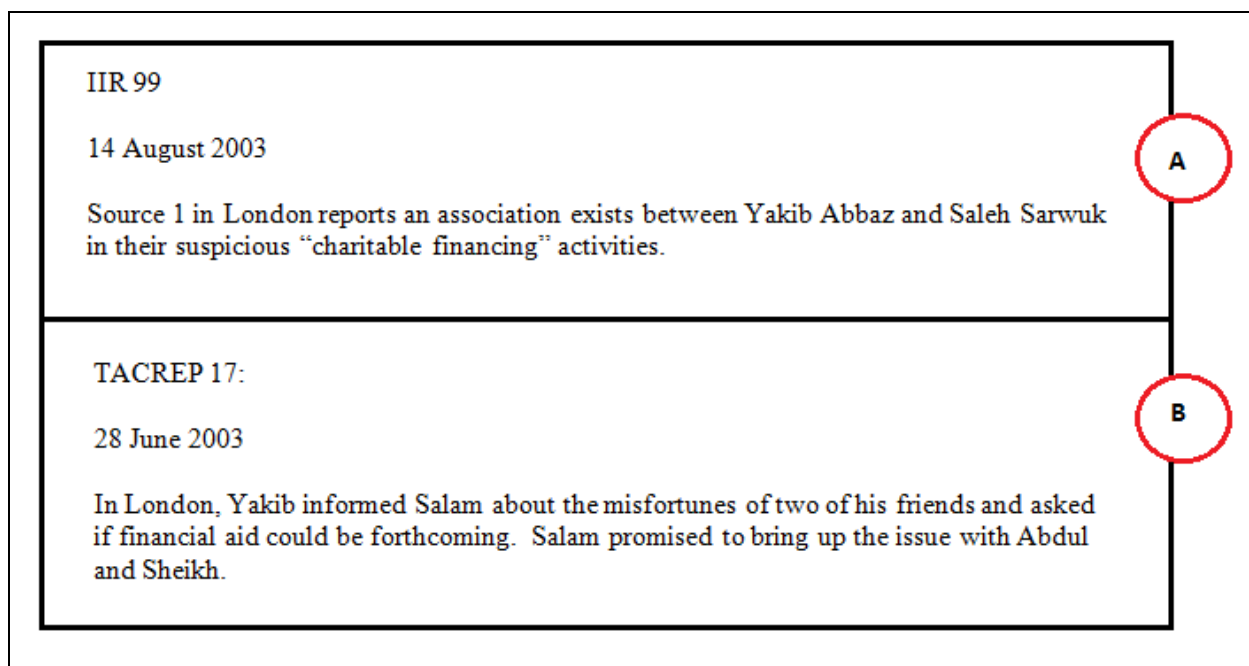


Figure 24. Corresponding messages from the ARL Ali Baba Data Set.

The increase in complexity from the previous two-node search query (section 3) to this three-node search query highlights the advantages of using a graph-based approach (i.e., visual graph interface to a subsubgraph matching algorithm) over more traditional text-based searching techniques. For example, multinode network searches cannot be easily formulated in text-box search interfaces offered by most text-based applications (e.g., Microsoft Word, Excel) and text-based search engines (e.g., Google). Complex multinode searches constructed in GMT’s canvas can be saved for reuse, or used as a starting point and extended or modified for searching other data sources. Saving search queries is explained in section 8.

8. Saving a Search

The messages reviewed by the analyst in figure 24A and 24B uncovered important information. The three-node search query that identified these messages can be saved to repeat the search as the Ali Baba Data Set is updated with new information or shared with other analysts. To save a search, the Save... is selected from the File option or the shortcut Ctrl+S is used (figure 25). Searches are saved as a text document. Saved searches can be reopened by selecting Open... from under the File option or the shortcut Ctrl + O (figure 26).

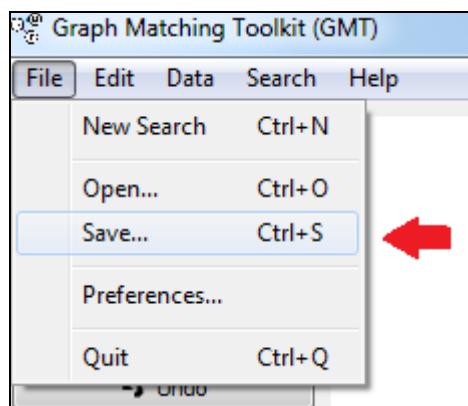


Figure 25. Saving searches.

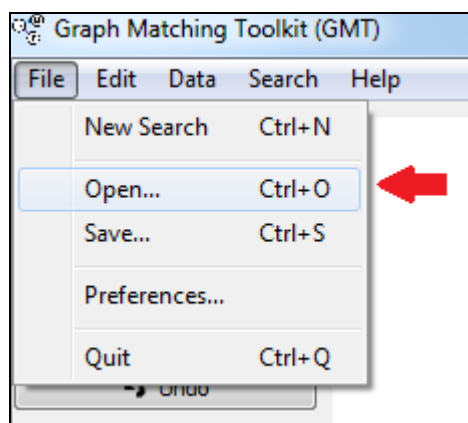


Figure 26. Opening saved searches.

9. Starting a New Search

If the GMT canvas contains a search query, the canvas can be cleared by selecting File from the menu bar at the top of the screen and then the first option New Search, or the shortcut Ctrl+N can also be used (figure 27).

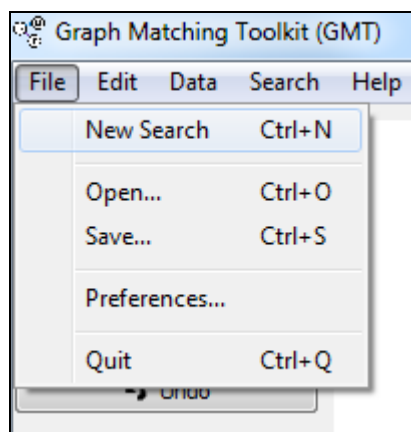


Figure 27. Starting a New Search.

10. Discussion

This report demonstrates the basic search functionality of GMT using example search queries in the context of intelligence gathering. Graph-based approaches such as GMT's visual graphical interface in combination with the TruST subgraph matching algorithm can assist intelligence analysts working in time-constrained environments where traditional text-based manual analysis of large data collections is impractical.

An example scenario from ARL's Ali Baba Data Set was used to illustrate step-by-step construction of network search queries consisting of nodes, links, and attributes (categories and labels) on GMT's canvas window using a mouse and drop-down menus. A description of how to execute a search including selecting the dataset and search algorithm was given. Guidelines for interpreting GMT search results were provided. These step-by-step instructions with accompanying screen shots should be used as an introduction for learning the GMT interface and the process of subgraph matching.

By using GMT to search the Ali Baba Data Set, the analyst in the example was able to perform a concise and effective network search in a timely manner. GMT's capabilities facilitate complex searches allowing more time for the analysis, distribution, and synthesis of the data with other multisource information that may be available (e.g., images, sensor data).

The appendixes attached to this report offer additional information such as a more detailed use case of GMT in an intelligence analysis context (appendix A), an installation guide (appendix B), and contact information for technical GMT support (appendix C).

11. Conclusions and Future Revisions

GMT allows analysts to iteratively search a large data graph, such as a social network extracted from social media by NLP tools, for matches to subgraphs constructed by the analysts. This capability enables analysts to quickly find high-value targets within large social networks without having to manually search through intelligence documents. Because GMT was designed to be general purpose, it can serve as an intuitive interface to many different graph matching algorithms.

Future work on GMT will include interfacing GMT with additional graph matching algorithms, and adding an option to the File menu to allow the analyst to load multiple saved searches into a single integrated diagram. Allowing the analyst to load many saved searches into a single integrated diagram will facilitate an overall view of the analyst's hypotheses; for example, who the high-value target communicates with, what the high-value target's activities are, and so forth. These additional features should improve the usability of GMT for intelligence analysts.

12. References

1. Sambhoos, K.; Nagi, R.; Sudit, M.; Stotz, A. Enhancements to High Level Data Fusion Using Graph Matching and State Space Search. *Information Fusion* **2010**, *11* (4), 351–364.
2. Qt Digia Home Page. Qt SDK. <http://qt.digia.com/Product/Qt-SDK> (accessed on 18 March 2013).
3. Sambhoos, K. Graph Matching Applications in High Level Information Fusion. Ph.D dissertation, State University of New York at Buffalo, 2007.
4. Sudit, M.; Nagi, R.; Stotz, A.; Sambhoos, K. A Graph-Based Framework for Fusion: From Hypothesis Generation to Forensics. In *9th International Conference on Information Fusion*; IEEE Press: New York, 2006.
5. Sambhoos, K.; Nagi, R.; Sudit, M.; Rickard, T. Hierarchical Higher Level Data Fusion Using Fuzzy Hamming and Hypercube Clustering. *Journal of Advances in Information Fusion* **2008**, *3* (2), 90–106.
6. Chen, S.; Chen, J. Fuzzy Risk Analysis Based on Ranking Generalized Fuzzy Numbers With Different Heights and Different Spreads. *Expert Systems With Applications* **2009**, *36* (3), 6833–6842.
7. Jaworowski, M.; Pavlak, S. *Ali Baba Data Set Ground Truth*; U.S. National Security Agency: Fort Meade, MD, 2003.
8. Mittrick, M.; Roy, H.; Kase, S.; Bowman, E. *Refinement of the Ali Baba Data Set*; ARL-TN-0467; U.S. Army Research Laboratory: Aberdeen Proving Ground, MD, 2012.

INTENTIONALLY LEFT BLANK.

Appendix A. Graph Matching Toolkit Use Case: Extended Ali Baba Example Query

This appendix introduces a scenario to provide a multistage search example of how the Graph Matching Toolkit (GMT) may be used by an intelligence analyst to search large networks of data for information pertaining to high-value targets. This example uses the U.S. Army Research Laboratory Ali Baba Data Set to conduct a three-stage search for information. As might occur in the field, the analyst uses AXIS Pro in addition to GMT to organize the gathered intelligence data. AXIS Pro is the primary social network analysis tool used by the U. S. Army.

To begin this scenario, the analyst receives a police report (figure A-1) that Yakib Abbaz is a suspected terrorist. The police request additional information about Yakib Abbaz and his relationships. Therefore, the analyst wants to search the data graph of text communications for other tactical reports (TacReps) that reference Yakib Abbaz.

Police Report 60 10 October 2003 London Police picked up an associate whom they believe to be an associate of suspected terrorist recruiter, Yakib Abbaz. The man, Khalid Kalhoun, is currently being detained and is reportedly not cooperating with investigators.
--

Figure A-1. Police Report 60.

To perform the search, the analyst draws a two-node subgraph query in GMT's canvas subwindow. The subgraph has node 1 (n1) as a Person labeled Yakib Abbaz and node 2 (n2) as a TacRep (Tactical Report) unlabeled (figure A-2, left side). The nodes are then connected by a Contained In link (figure A-2, light blue arrow) that indicates Yakib Abbaz's name may appear in other reports.

Once the subgraph is drawn, the analyst executes the graph matching algorithm by selecting the Truncated Search Tree (TruST) algorithm from the menu. The results of the search return the top five scoring matches (figure A-2, right side). Those matches indicate specific TacReps that contained information about Yakib Abbaz. The score of 0.9 for each match indicate a nearly perfect match was found to the subgraph. The analyst then examines the specific TacReps that correspond to those top five matches in greater detail.

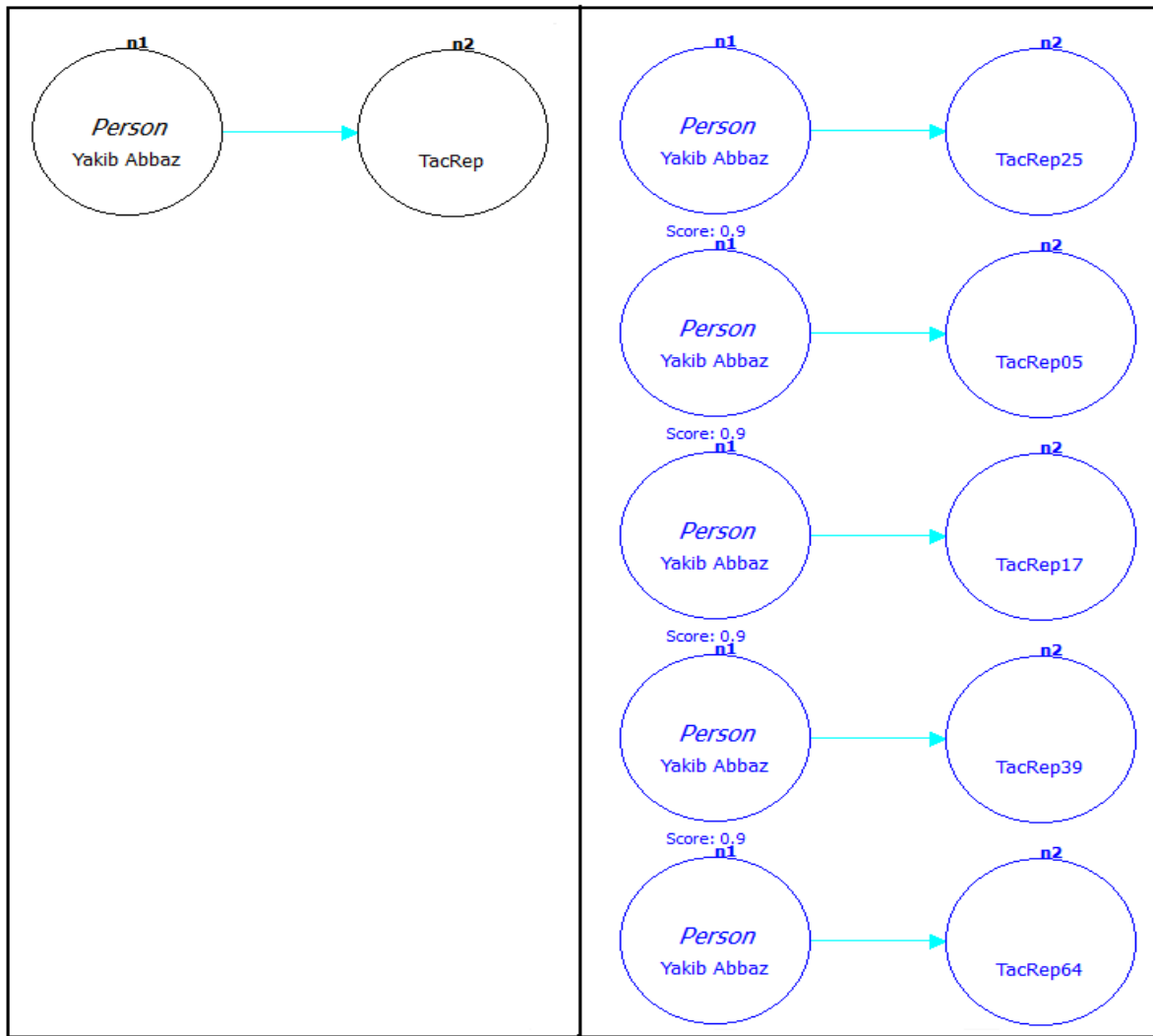


Figure A-2. Two-node template graph for a search (left side) and search results (right side).

From the text contained in the TacReps listed in the search results, more information is gleaned about Yakib Abbaz. Yakib communicated with Salam Seeweel, who appears to be a financier, for an operation referred to as “baking a cake” (figure A-3, TacRep 25). Also, Yakib may be recruiting “apprentice bakers” at the East Side Mosque (figure A-3, TacReps 5 and 39). In TacRep 4, Yakib reports his success to Imad Abdul. And in TacRep 64, Yakib is “ordered” by Abdul. Thus, Imad Abdul appears to be acting in a leadership capacity at a level above Yakib.

TACREP 25 14 July 2003 In London, Salam Seeweel informed Yakib Abbaz that money would be available for all who would assist in baking a big, delicious cake. Money would also be available for the newcomers to the party. Yakib expressed his gratitude.
TACREP 5 06 June 2003 In London, Yakib Abbaz reported to Imad Abdul that he (Yakib) was able to identify several high-quality prospects at the East Side Mosque. Abdul praised Yakib for his good work.
TACREP 17 28 June 2003 In London, Yakib informed Salam about the misfortunes of two of his friends and asked if financial aid could be forthcoming. Salam promised to bring up the issue with Abdul and Sheikh.
TACREP 39 19 August 2003 In London, Raed told Tarik that Yakib Abbaz had done a very good job in bringing apprentice bakers on the mission.
TACREP 64 14 October 2003 In the town of Henley, Abdul told ordered Yakib to cease his activity in London and to take a vacation in the northern part of England.

Figure A-3. The messages from the dataset corresponding to the search results.

With the new information gathered from the TacReps, the analyst characterizes Yakib Abbaz's relationships using a concept map. The TacReps are imported into a social network analysis tool, AXIS Pro, for further analysis. Figure A-4 shows an AXIS Pro concept map of Yakib Abbaz's relationships derived from the GMT search results.

Because Imad Abdul appears to be high in the chain of command and the analyst is interested in Yakib Abbaz's social network, the next step is to use GMT to search the database for communications involving both Imad Abdul and Yakib Abbaz. A three-node subgraph is constructed with node 1 (n1) as a Person labeled Imad Abdul, and node 2 (n2) as a Person labeled Yakib Abbaz, connected with a Contains Person link to node 3 (n3) as a TacRep (figure A-5). When the search is executed, GMT displays the top five scoring matches in the form of TacRep numbers. These specific TacReps are then pulled from the database for a more in-depth analysis, and the concept map is expanded with the additional information.

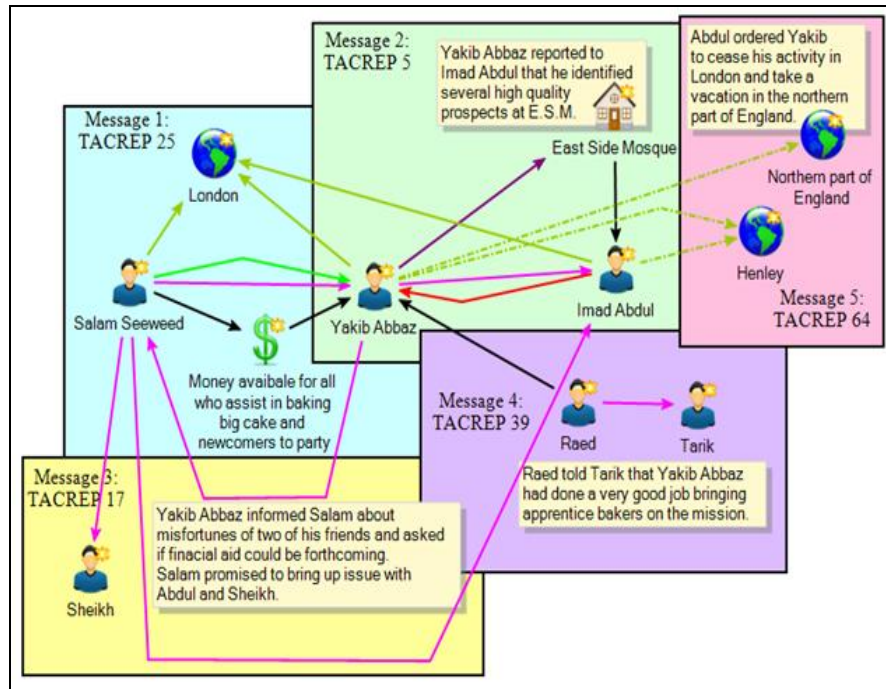


Figure A-4. AXIS Pro concept map of GMT search results outlined by TacRep number.

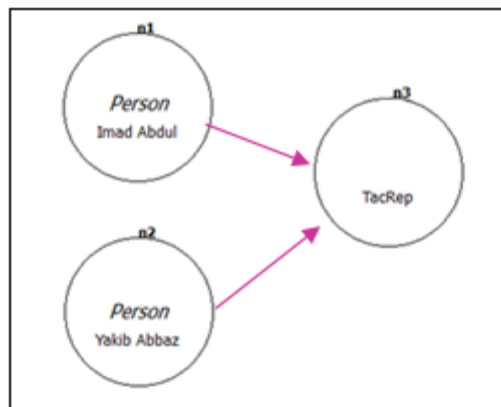


Figure A-5. GMT three-node search template.

Results from the three-node subgraph search reveal five matches with a score of 0.9. Two of the five subgraph matches are new TacReps; the other three were contained in the previous search results. The information gained from the new TacRep messages may be integrated into the analyst's concept map. The first new message, TacRep1, provides further evidence that Abdul is higher in the chain of command than Yakib by stating "Imad tasked Yakib." In this message, Yakib is tasked to visit several mosques in eastern London to identify persons sympathetic with their cause. In the second new message, TacRep 19, Yakib reports back to Abdul that the response at the Sheepside Mosque in western London was less receptive than eastern London. Together, this information substantiates Yakib's role as a recruiter.

GMT search results can be adjusted to show the top 5–10 scored matches. When the result set is expanded to 10, the sixth message (TacRep 50) scoring 0.8 is a “fuzzy” match, which introduces a new person, Ali Baba. The text contained in TacRep 50 indicates Ali Baba and Imad Abdul are connected by a “cake baking” operation (figure A-6) and that Ali Baba may be higher in the chain of command.

TACREP 50 29 September 2003 In the town of Henley, Abdul reported to Ali Baba that all preparations for baking the cake were proceeding on schedule.
--

Figure A-6. GMT fuzzy match result, TacRep 50

With this new information, a power hierarchy of direct relationships within the social network began to emerge. Figure A-7 demonstrates how AXIS Pro is used to create a link diagram that integrates and displays the new information. At this point, the analyst wants to search the database for communications involving Ali Baba. This leads to the discovery that Ali Baba is the leader of a terrorist organization called the Ali Baba Group, which is planning to bomb a water treatment facility, referred to as “baking a cake.”

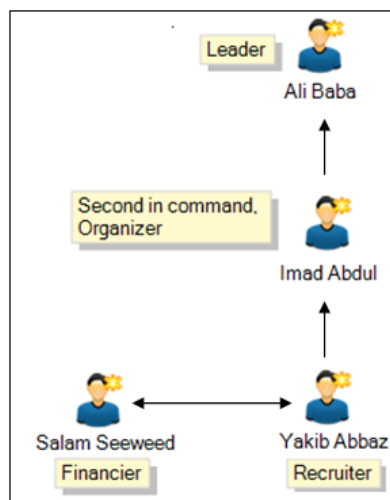


Figure A-7. AXIS Pro link diagram of the power hierarchy of the Ali Baba Group.

Appendix B. Graph Matching Tool Installation Guide

B.1 Overview

The Graph Matching Toolkit (GMT) provides a graphical interface that employs graph matching algorithms such as the Truncated Search Tree (TruST) graph matching algorithm discussed in the body of this report, to discover relevant matches that facilitate an intelligence analyst's analysis of critical social networks. Graph matching algorithms allow the user to search a graph database for matches to user-specified graph patterns. GMT can be installed on either the Windows or UNIX operating system platforms on a virtual or real machine.

The following files are common to installations on either the Windows or UNIX operating system platforms:

- configuration file (*gmt.conf*)
- data sources file (*data-sources.txt*)
- node/link ontology files:
 - ARL/Orbis/MURI Ali Baba ontology files
 - ARL/Orbis SYNCOIN ontology files
 - TIFB ontology files

The configuration file (*gmt.conf*) for GMT follows the standard format for UNIX configuration files. Each line of text specifies the name of a field and its value. See table B-1 for an example of how the names and the descriptions of the fields look when using the TruST algorithm.

Table B-1. The description of the fields found in the GMT configuration file.

Field Name	Description
trust-program	The path to the TruST executable file.
trust-template-graph	The path to use for the TruST template graph file.
trust-output-file	The path to use for the TruST output file.
trust-node-scores-file	The path to use for the TruST node scores file.
trust-edge-scores-file	The path to use for the TruST edge scores file.
trust-increment-directory	The path to use for the TruST increment directory.
trust-increment-scores-directory	The path to use for the TruST increment scores directory.
trust-score-threshold	The value of the score threshold parameter to pass to the TruST executable.
trust-parm-power-mean	The value of the power mean parameter to pass to the TruST executable.
trust-parm-root-weight	The value of the root weight parameter to pass to the TruST executable.
trust-parm-ko	The value of the Ko parameter to pass to the TruST executable.
trust-parm-ki	The value of the Ki parameter to pass to the TruST executable.
trust-parm-beta	The value of the beta parameter to pass to the TruST executable.
trust-parm-depth	The value of the depth parameter to pass to the TruST executable.
trust-scorer	The command line to use to run the TruST scorer.
iq-program	The path to the script which runs the IQ algorithm.
iq-query-filename	The path to use for the query file for the IQ algorithm script.
iq-constraint-filename	The path to use for the constraint file for the IQ algorithm script.
iq-score-filename	The path to use for the score file for the IQ algorithm script.
iq-query-id	The value of the query ID parameter to pass to the IQ algorithm script.
iq-max-nr-results	The value of the maximum number of results parameter to pass to the IQ algorithm script.
iq-algorithm-id	The value of the algorithm ID parameter to pass to the IQ algorithm script.
iq-database-directory	The path to use for the database directory for the IQ algorithm script.
iq-misc-parms	The values of the miscellaneous parameters to pass to the IQ algorithm script.
data-sources-filename	The path to the file containing the possible data sources.
current-data-source	The human readable name for the current data source.
max-nr-search-results	The maximum number of search results to display.

B.2 Windows Installation

GMT can be easily installed on the Windows operating system platform using GMT's installation wizard. Start the installation wizard by double-clicking on the icon and follow the instructions to install GMT (see figures B-1 through B-4).

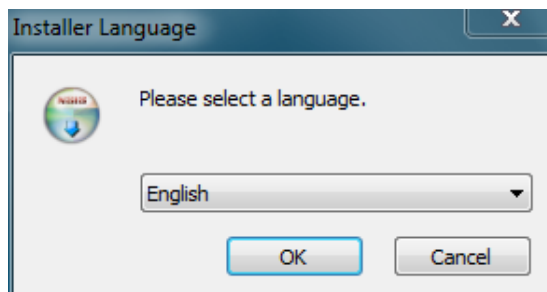


Figure B-1. The first step for the GMT installation wizard.

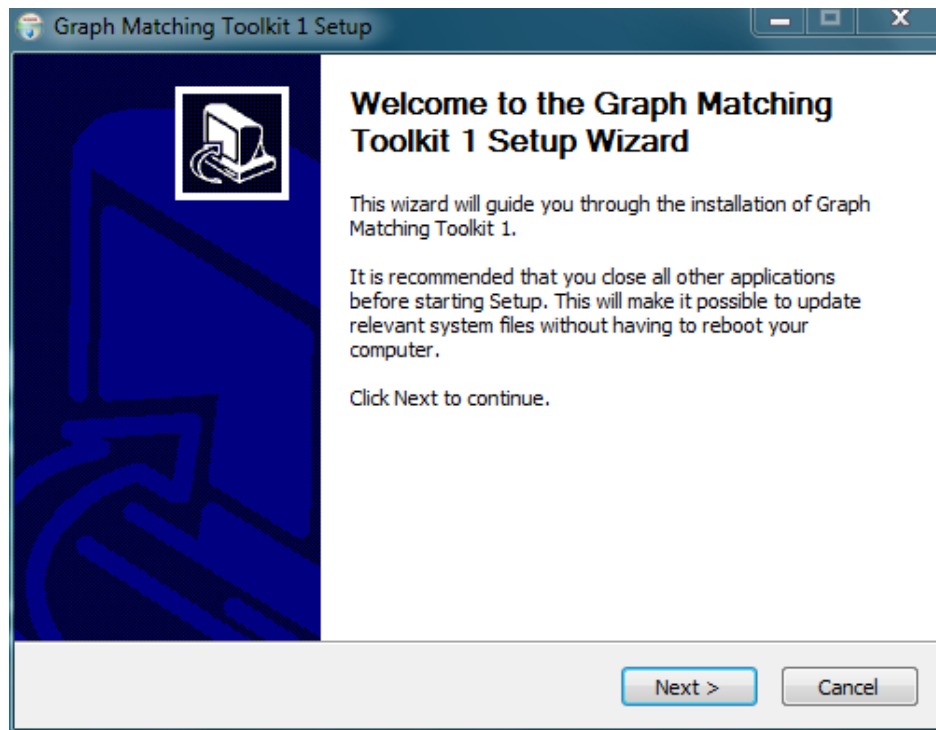


Figure B-2. The second step for the GMT installation wizard.

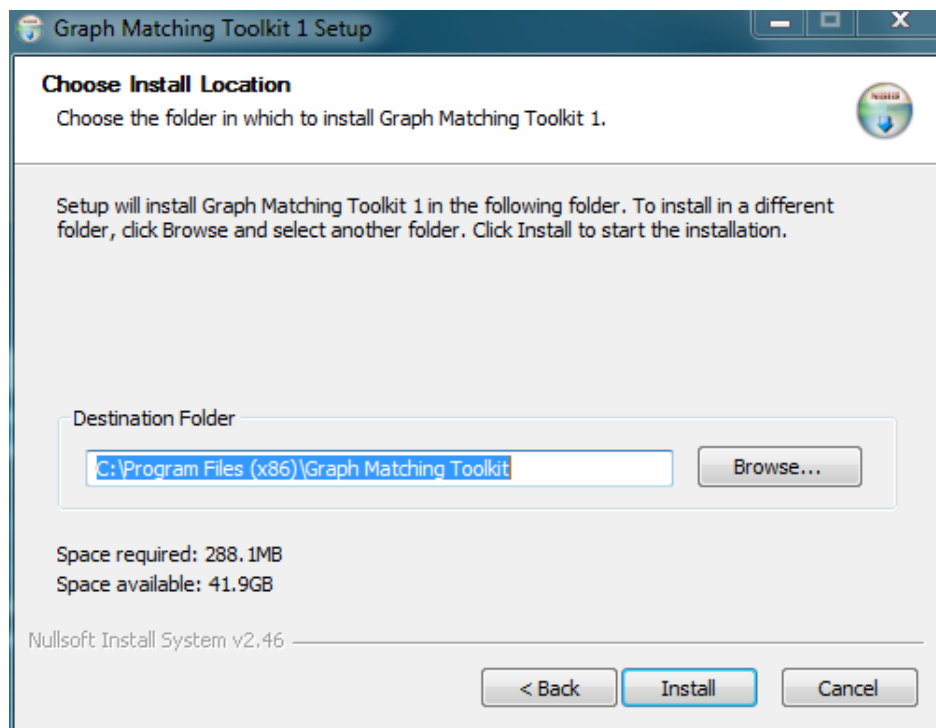


Figure B-3. The third step for the installation wizard.

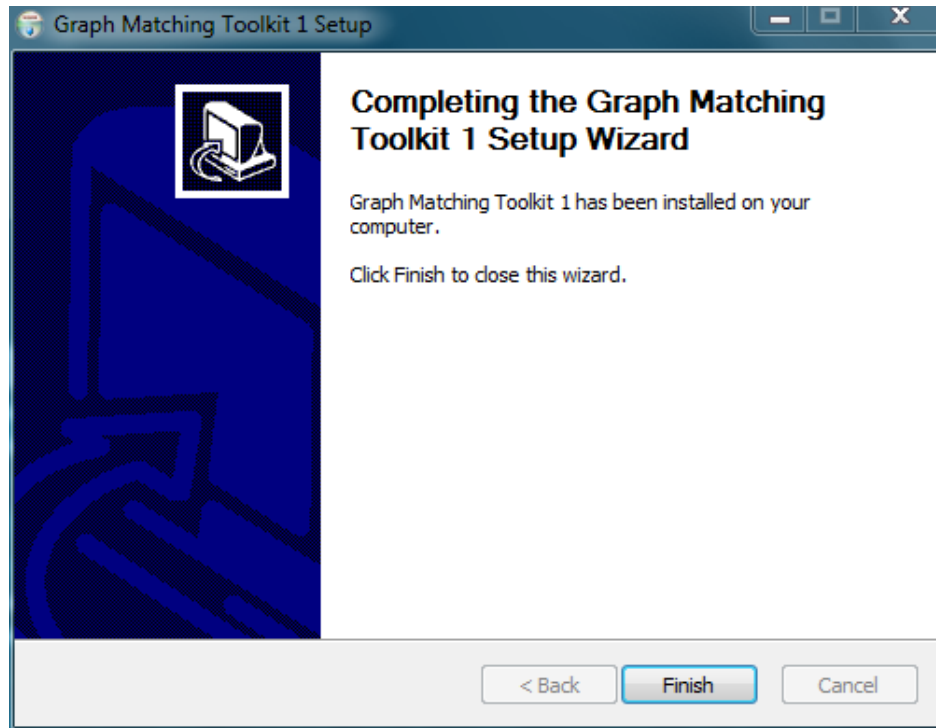


Figure B-4. The fourth step for the installation wizard.

B.3 UNIX Installation

Unlike the Windows operating system platform, there is no installation script to facilitate the installation of GMT on the UNIX operating system platform. Thus, UNIX installation is a more complicated procedure. First, ensure that the correct libraries are installed on the computer. GMT requires the open-source Qt development libraries (found at <http://qt-project.org>). Once the Qt development libraries are installed, compile the source code using the *make* command. The source code distribution for GMT contains all the necessary files for running GMT. The compiled executable for GMT can be started directly from the source directory. However, the default values in the configuration file (*gmt.conf*) will need to be adjusted first. Specifically, the directories provided in the configuration file for the various files needed by GMT will need to be changed to the installation directory.

INTENTIONALLY LEFT BLANK.

Appendix C. Graph Matching Tool Support Contact Information

Please contact Dr. Kirk Ogaard with any questions, concerns, or feedback regarding the use of Graph Matching Toolkit (GMT). Requests for the GMT software should also be directed to Dr. Ogaard, who is the creator of GMT and a postdoctoral researcher for the Computational and Information Sciences Directorate at the U.S. Army Research Laboratory. Dr. Ogaard may be contacted at: <kirk.a.ogaard.ctr@mail.mil>.

NO. OF
COPIES ORGANIZATION

1 (PDF)	DEFENSE TECHNICAL INFORMATION CTR DTIC OCA
1 (PDF)	DIRECTOR US ARMY RESEARCH LAB IMAL HRA
1 (PDF)	DIRECTOR US ARMY RESEARCH LAB RDRL CIO LL
1 (PDF)	GOVT PRINTG OFC A MALHOTRA
1 (PDF)	ARMY G2 D WALSH
1 (PDF)	MULTISOURCE INFO FUSION RESEARCH PROFESSOR (EMERITUS) J LLINAS
17 (PDF)	US ARMY RSRCH LAB RDRL CII B BROOME M THOMAS RDRL CII A S H YOUNG RDRL CII B L TOKARCIK R WINKLER RDRL CII C B BODT E K BOWMAN F S BRUNDICK J DUMER T HANRATTY E HEILMAN S KASE M MITTRICK A NEIDERER J RICHARDSON H ROY RDRL CII T V M HOLLAND
1 (PDF)	US ARMY CERDEC I2WD RDER IWP D PORTER

INTENTIONALLY LEFT BLANK.