Matrix to Coordinates (M2C) 2.1 Documentation

Benjamin Elbirt

Table of Contents

Software License Agreement

PLEASE READ THIS SOFTWARE LICENSE AGREEMENT CAREFULLY BEFORE INSTALLING OR USING THE SOFTWARE.

BY OPENING THE PACKAGE, DOWNLOADING THE PRODUCT, OR USING THE EQUIPMENT THAT CONTAINS THIS PRODUCT, YOU ARE CONSENTING TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF THIS AGREEMENT, RETURN THE PRODUCT TO THE PLACE OF PURCHASE FOR A FULL REFUND, OR DO NOT DOWNLOAD THE PRODUCT.

Single User License Grant: Benjamin Elbirt (“Owner”) grants to Customer ("Customer") a nonexclusive and nontransferable license to use **Matrix to Coordinates 2.1 ©/M2C 2.1©** ("Software") in object code form solely on a single central processing unit owned or leased by Customer.

Multiple-Users License Grant: Benjamin Elbirt (“Owner”) grants to Customer ("Customer") a nonexclusive and nontransferable license to use **Matrix to Coordinates 2.1 ©/M2C 2.1** ("Software") in object code form: (i) installed in a single location on a hard disk or other storage device of up to the number of computers owned or leased by Customer for which Customer has paid a license fee ("Permitted Number of Computers"); or (ii) provided the Software is configured for network use, installed on a single file server for use on a single local area network for either (but not both) of the following purposes: (a) permanent installation onto a hard disk or other storage device of up to the Permitted Number of Computers; or (b) use of the Software over such network, provided the number of computers connected to the server does not exceed the Permitted Number of Computers. Customer may only use the programs contained in the Software (i) for which Customer has paid a license fee (or in the case of an evaluation copy, those programs Customer is authorized to evaluate. Customer grants to Owner or its independent accountants the right to examine its books, records and accounts during Customer's normal business hours to verify compliance with the above provisions. In the event such audit discloses that the Permitted Number of Computers is exceeded, Customer shall promptly pay to Owner the appropriate licensee fee for the additional computers or users. At Owner option, Owner may terminate this license for failure to pay the required license fee.

Customer may make one (1) archival copy of the Software provided Customer affixes to such copy all copyright, confidentiality, and proprietary notices that appear on the original.

EXCEPT AS EXPRESSLY AUTHORIZED ABOVE, CUSTOMER SHALL NOT: COPY, IN WHOLE OR IN PART, SOFTWARE OR DOCUMENTATION; MODIFY THE SOFTWARE; REVERSE COMPILE OR REVERSE ASSEMBLE ALL OR ANY PORTION OF THE SOFTWARE; OR RENT, LEASE, DISTRIBUTE, SELL, OR CREATE DERIVATIVE WORKS OF THE SOFTWARE.

Customer agrees that aspects of the licensed materials, including the specific design and structure of individual programs, constitute trade secrets and/or copyrighted material of Owner. Customer agrees not to disclose, provide, or otherwise make available such trade secrets or copyrighted material in any form to any third party without the prior written consent of Owner. Customer agrees to implement reasonable security measures to protect such trade secrets and copyrighted material. Title to Software and documentation shall remain solely with Owner.

LIMITED WARRANTY. Owner warrants that for a period of ninety (90) days from the date of shipment from Owner: (i) the media on which the Software is furnished will be free of defects in materials and workmanship under normal use; and (ii) the Software substantially conforms to its published specifications. Except for the foregoing, the Software is provided AS IS. This limited warranty extends only to Customer as the original licensee. Customer's exclusive remedy and the entire liability of Owner under this limited warranty will be, at Owner option, repair, replacement, or refund of the Software if reported (or, upon request, returned) to the party supplying the Software to Customer. In no event does Owner warrant that the Software is error free or that Customer will be able to operate the Software without problems or interruptions.

This warranty does not apply if the software (a) has been altered, except by Owner, (b) has not been installed, operated, repaired, or maintained in accordance with instructions supplied by Owner, (c) has been subjected to abnormal physical or electrical stress, misuse, negligence, or accident, or (d) is used in ultra-hazardous activities.

DISCLAIMER. EXCEPT AS SPECIFIED IN THIS WARRANTY, ALL EXPRESS OR IMPLIED CONDITIONS, REPRESENTATIONS, AND WARRANTIES INCLUDING, WITHOUT LIMITATION, ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE, ARE HEREBY EXCLUDED TO THE EXTENT ALLOWED BY APPLICABLE LAW.

IN NO EVENT WILL Owner BE LIABLE FOR ANY LOST REVENUE, PROFIT, OR DATA, OR FOR SPECIAL, INDIRECT, CONSEQUENTIAL, INCIDENTAL, OR PUNITIVE DAMAGES HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY ARISING OUT OF THE USE OF OR INABILITY TO USE THE SOFTWARE EVEN IF Owner HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. In no event shall Owner liability to Customer, whether in contract, tort (including negligence), or otherwise, exceed the price paid by Customer. The foregoing limitations shall apply even if the above-stated warranty fails of its essential purpose. SOME STATES DO NOT ALLOW LIMITATION OR EXCLUSION OF LIABILITY FOR CONSEQUENTIAL OR INCIDENTAL DAMAGES.

The above warranty DOES NOT apply to any beta software, any software made available for testing or demonstration purposes, any temporary software modules or any software for which Owner does not receive a license fee. All such software products are provided AS IS without any warranty whatsoever.

This License is effective until terminated. Customer may terminate this License at any time by destroying all copies of Software including any documentation. This License will terminate immediately without notice from Owner if Customer fails to comply with any provision of this License. Upon termination, Customer must destroy all copies of Software.

Software, including technical data, is subject to U.S. export control laws, including the U.S. Export Administration Act and its associated regulations, and may be subject to export or import regulations in other countries. Customer agrees to comply strictly with all such regulations and acknowledges that it has the responsibility to obtain licenses to export, re-export, or import Software.

This License shall be governed by and construed in accordance with the laws of the State of New York, United States of America, as if performed wholly within the state and without giving effect to the principles of conflict of law. If any portion hereof is found to be void or unenforceable, the remaining provisions of this License shall remain in full force and effect. This License constitutes the entire License between the parties with respect to the use of the Software.

I M2C 2.1 Product Specification

I Product Overview

M2C 2.1 © is an application written in Java that translates matrixes of varying size into Riemann coordinate systems. These coordinate systems can be used for displaying the relationship between objects within the matrix as a function of distance.

This application is not a visualization tool and does not provide a means of visualizing the resulting coordinates. Alternative applications are necessary for visualization as follows:

Sample Visualization Tools

1. Jacob's Ladder© - http://www.elbirttechnologies.com
2. ThoughtView™ - http://www.galileoco.com
3. NetDraw™ - http://www.analytictech.com/Netdraw/netdraw.htm
4. MultiNet™ - http://www.sfu.ca/~richards/Multinet/Pages/multinet.htm
5. MS Excel™ - http://office.microsoft.com/en-us/excel/default.aspx

II Software Installation

M2C 2.1© requires the installation of the latest Java environment which is available at http://www.java.com/en/download/index.jsp.

The file *m2c.jar*, which represents the **M2C 2.1**© executable Java application, can be placed anywhere within the file system and will run in any window environment (X-Windows, MS Windows, Mac OS) that supports Java.

III Memory Expansion and Consumption

M2C 2.1© uses as much memory as made available through the Java environment and the command line execution. To increase the memory, use:

```
java –jar –Xms 1M  –Xmx 2M m2c.jar
```

where ***1M*** is the amount of memory to use for the initial heap size and ***2M*** is the amount of memory to use for the maximum heap size. The **M** is used to represent megabytes.

```
java –jar –Xms 1000M –Xmx 2000M m2c.jar
```

will launch the application with an initial heap of 1 GB and a maximum heap of 2 GB.

IV Bug Collection & Reporting

Use the "java –jar m2c.jar" command from a prompt (DOS/Unix/Mac) to obtain error messages from the executable. Please report any bugs / problems to the author, Benjamin Elbirt, at sarq_6669@yahoo.com.

V Future Enhancements

The following enhancements are planned for future releases:

1. Excel File Inputs
2. Rotation Algorithm
3. GraphML Input
4. Additional Output Formats

II M2C 2.1 Theory

M2C 2.1© is based on general network theory (Wasserman & Faust, 1994), uses Riemann space MDS methodology (Brandes & Pich, 2007; Woelfel & Fink, 1980), and a custom rotation algorithm was built using Wikipedia – Rotation (mathematics) and related links.

I Networks

"Networks" is a philosophical approach to the study of anything that can be represented by one or more matrixes of relations. Network theory/methods using matrixes of relations exist for social, political, technological and other areas of study.

The approach specifically considers the relationship between the various objects within a system. For example, a matrix of relations can be created that represents the number of e-mail exchanges that occur between individuals in a given period of time. This matrix would be considered a "Strength" matrix because the values represent a count of e-mails sent.

Various calculations can be made on this matrix (often called a network); however this application is not for the study of networks. Rather, it is for the manipulation of networks into Riemann geometry spatial systems for data visualization.

II Riemann vs. Euclidean Geometry

MDS algorithms exist for creating Euclidean geometries; however not all networks (especially cognitive networks) neatly fit within a standard Euclidean geometry (Woelfel & Fink, 1980). Riemann geometry is the accepted alternative for representing non-Euclidean geometries (Woelfel & Fink, 1980; Brandes & Pich, 2007).

M2C 2.1© specifically uses Riemann geometry to ensure all networks provided are properly converted into coordinate systems.

III Strength to Distance Conversion

M2C 2.1© converts all strength matrixes to distance matrixes prior to generation of coordinates. Two conversion processes are provided – Flip and Structural Equivalence (SE).

Flip Conversion

The conversion process first finds the largest absolute value in the matrix. It then converts all values as Value = (Max – Original Value). This has the effect of making the largest value zero and the smallest value the largest value.

This process applies to all matrix cells except the diagonal which is always zero. Zero diagonals satisfies the condition that you cannot be distant from yourself.

Structural Equivalence (SE) Conversion

This method compares the object rows/columns to each other to determine how different they are. The difference is calculated using the Pythagorean Theorem such that the distance between two nodes is the square root of the sum of all differences squared.

The final matrix is a symmetrical matrix of distances representing the distance between objects based on the differences between their rows and columns in the original matrix.

$$diff(i,j) \quad = \quad diff(j,i) \quad = \quad \sqrt{\sum_{k=1}^{n}\left(((i,k)-(j,k))^2+((k,i)-(k,j))^2\right)}$$

IV Coordinates (Multi-Dimensional Scaling)

MDS (Multi-dimensional scaling) is the methodology used to convert distance matrixes to coordinate systems and has been used by many software packages across multiple disciplines. This software uses the Brandes & Pich (2007) java code exclusively based on the following assessments.

A comparison of the Woelfel & Fink (1980) methods with the Brandes & Pich (2007) algorithms yielded nearly identical results with the differences being less than 0.0001. This difference can be attributed to the use of Fortran (Woelfel & Fink) with four decimal precision vs. Java (Brandes & Pich) that has up to 11 decimal point precision.

A major difference between the two methods is the ability for the Brandes & Pich method to produce only the desired dimensionality where as the Woelfel & Fink method requires the coordinate system generated to be the same size as the matrix inputs. This significantly reduces processing time for solutions when a subset of the total dimensionality is needed.

Another important difference between the two software packages is the unlimited capabilities of the Brandes & Pich java packages (limited by memory/cpu available on the computer) vs. the Woelfel & Fink software that is limited to 999 concepts. These packages allow for the expansion of the software capabilities without code modifications.

Finally, the Brandes & Pich java packages are integrated into other java code where as the Woelfel & Fink software is an entire package that cannot be customized or integrated with other code. Although the Woelfel & Fink software (Galileo/CatPac) provides additional functionality/computations that the Brandes & Pich packages do not; these calculations can be added to future versions of Brandes & Pich where as the Woelfel & Fink software has not seen a significant update since the early 1990's.

V Comparing Spaces & Rotation

The algorithm use by **M2C 2.1©** to rotate coordinates has been removed due to errors.

III Skip the Theory; Let's Play! (Examples)

Although the theory is important to understanding how and why **M2C 2.1** © works, the theory is not necessary for application. The following is a partial list of applications for **M2C 2.1** © and examples of the usage.

Example 1 – Spatial Analysis (Driving Distance between US Cities)

This example uses the physical distances between major US cities as the input distance matrix. The figure below is a JL-12© visualization snapshot of the resulting display using the coordinates generated by **M2C 2.1**©. Although the results are spatially representative of the geographic location of the cities to each other; there is some discrepancy as the distances do not take into account the actual curvature of the earth (pear shaped). Each city is colored based on the voting party (democrat/republican blue/red) for the state in the 2008 Presidential election.

The software provides the Distance Matrix used with **M2C 2.1** to generate coordinates (us_cities.xls) and the NetXML attribute file (us_cities.netxml.txt) for creating the JL12 display XML file (us_cities.jl12.xml).

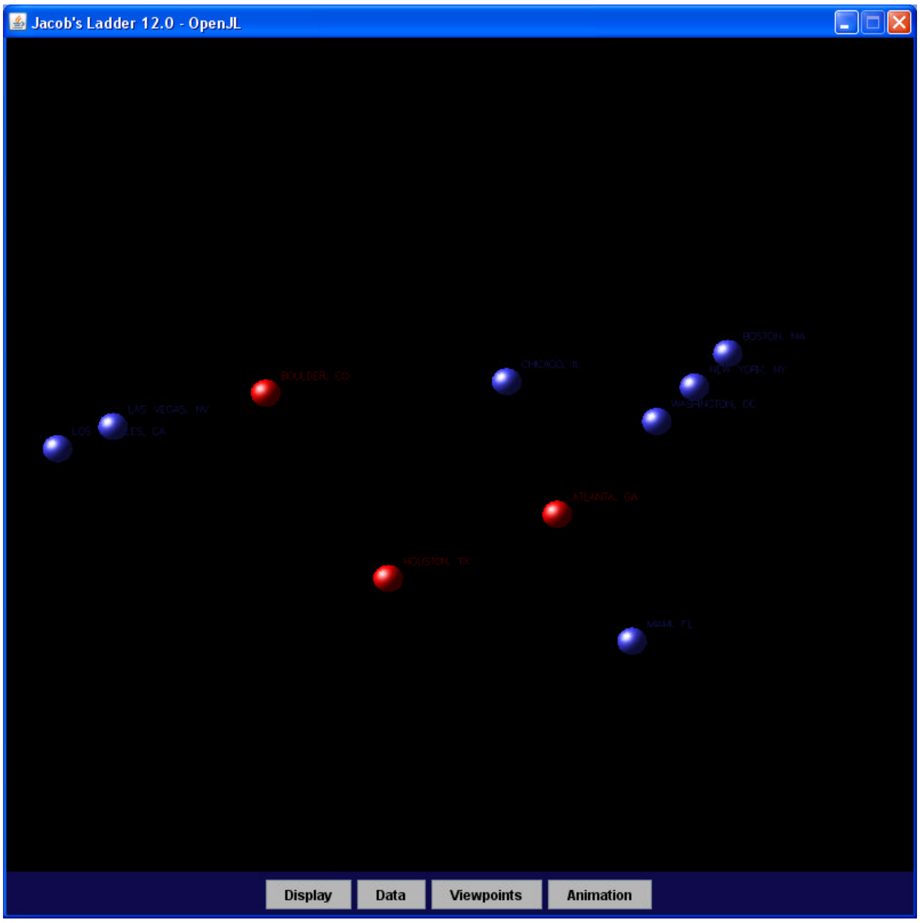


Diagram 1 – JL 12.1© Representation of US Cities Data

Example 2 – Semantic Analysis (English Version of Genesis)

This example uses the English version of Genesis as the data source for a semantic analysis. The resulting co-occurrence matrix of words (genesis.mx.txt) was run through **M2C 2.1©** to generate the coordinates (genesis.netxml.txt). Included in the coordinate file are the labels, row sums and related information for use with JL12©.

The diagram below provides a visualization of the genesis data where the closer words are the more related they are. The size of the object is a function of the number of times the word appears in the text; color has no meaning.

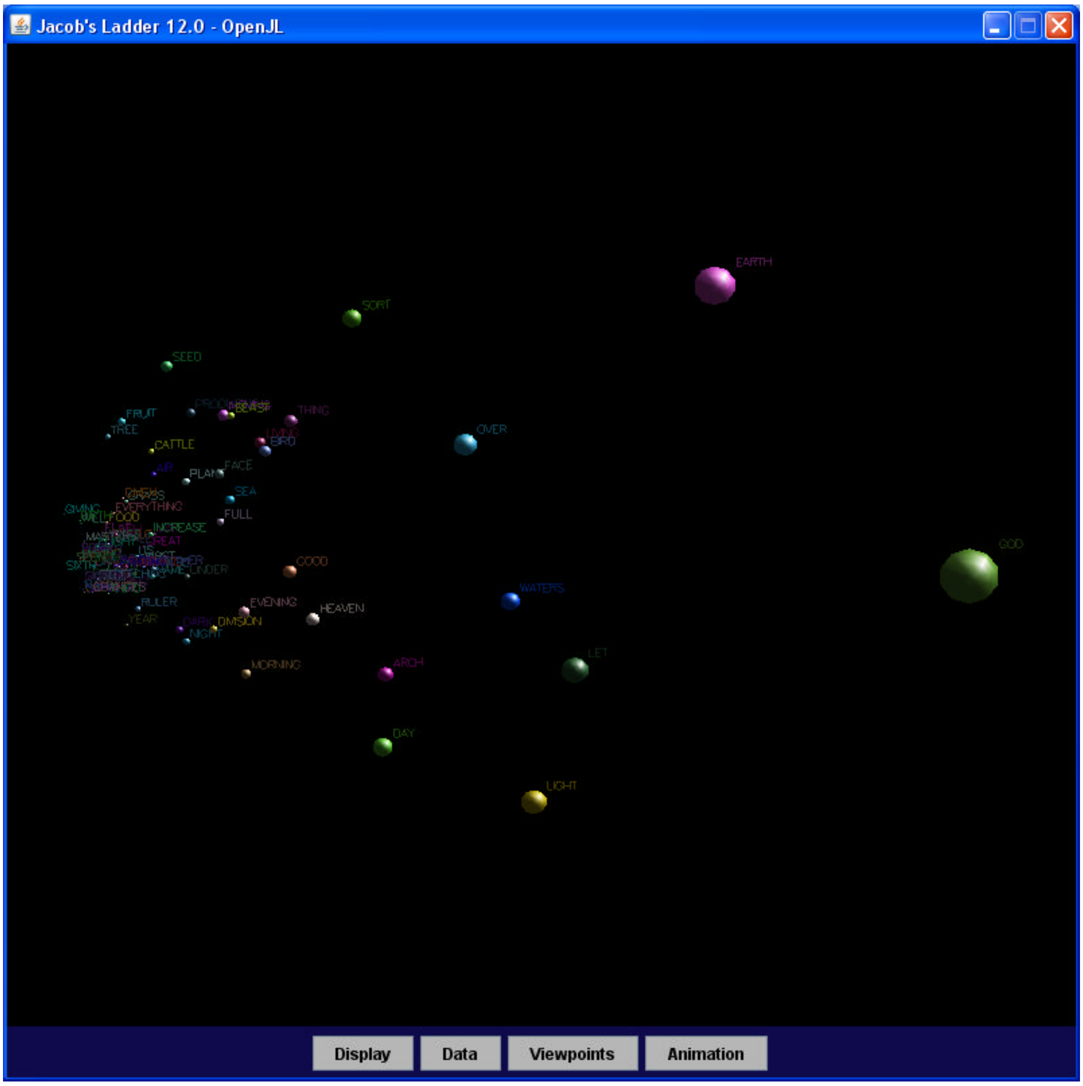
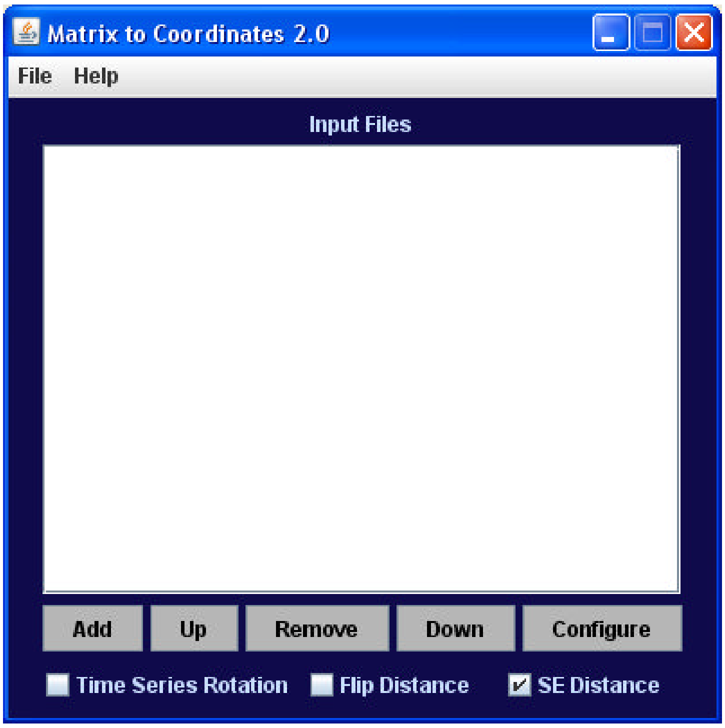


Diagram 2 – Genesis Semantic Analysis Coordinate Results

IV M2C 2.1 Application Interface



☐ Time Series Rotation ☐ Flip Distance ☑ SE Distance

Diagram 3 - M2C 2.1© Primary Interface

I File Menu

Save File Definitions

This option asks for a file name from the files system to save the current definitions to. A new file will be created if the one specified does not exist.

Resulting definitions are in UTF8 compatible format. Modification can be done external to **M2C 2.1©** at the risk of the user. Please see the section on Definition Output Files for more information of the format used.

Clear File Definitions

This option will clear all definitions currently stored within memory.

Load File Definitions

This option will ask for a file to load definition information from. An error message will be provided with detailed information regarding the error should a problem occur.

Process File Definitions

This option will process the specified files based on related configurations. An error message will be provided with detailed information should a problem occur during processing. The process will halt on any error.

Resulting coordinate files are produced after each input file is processed. Errors will not prevent previously processed output files from being generated and all output files will be over-written should they already exist.

II Help Menu

The help menu provides quick blurbs about the application inputs, outputs and history. This manual is a better source for information.

III Input File List

This list box contains the input files for processing. *Up*, *down*, *add, configure* and *remove* buttons provide the following controls regarding the input file list.

Up

This button will move the selected files up one space in the list. Order is only important when performing a time series rotation.

Down

This button will move the selected files down one space in the list. Order is only important when performing a time series rotation.

Add

This button provides a file specification interface and will add to the master list the file specified. Multiple files can be selected at once for addition.

Configure

This button will provide a configuration window for all selected input files in the input file list. Please see configuration information provided below.

Remove

This button will cause the removal of all selected files from the input file list.

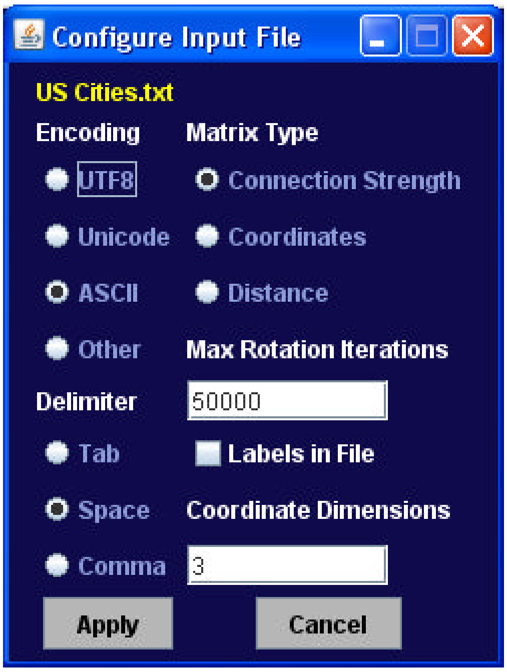
IV Time Series Rotation Option

This option has been removed from the interface until a valid algorithm can be implemented.

V Strength to Distance Conversion Options

Two checkboxes are provided for the method of strength to distance matrix conversion (Flip vs. Structural Equivalence).

VI Input File Configuration



US Cities.txt

Encoding
- ○ UTF8
- ○ Unicode
- ◉ ASCII
- ○ Other

Matrix Type
- ◉ Connection Strength
- ○ Coordinates
- ○ Distance

Max Rotation Iterations: 50000

Delimiter
- ○ Tab
- ◉ Space
- ○ Comma

☐ Labels in File

Coordinate Dimensions: 3

Diagram 4 - File Configuration Window

This input window allows the specification of parameters for creating the related coordinate file as follows:

Delimiter

This is the type of delimiter used in the matrix file. Options include tab, space and comma.

Encoding

This is the type of input file used allowing for multi-lingual input files. *Unicode* is the standard Java UTF16 format; *UTF8* is the standard UTF8 Unicode standard, *ASCII* is the standard ASCII character set and *Other* should be used for unknown file types.

Strength Matrix vs. Distance Matrix vs. Coordinate Matrix

This allows for determination of the type of matrix being input. A strength matrix will automatically be converted to a distance matrix. Please see the conversion theory section for more information on how this is done. Coordinate matrixes are only used for rotation sequences; the strength/distance matrix will not be calculated.

Rotation Iterations

This is an integer value specifying how much iteration to perform when trying to find best fit rotations. A limit on iteration volume is set by specifying an integer greater than one thousand (RI > 1000). This value will be ignored until a valid algorithm is implemented.

Coordinate Dimensions

This is an integer value of the number of coordinate dimensions to generate from the matrix data. A limit on dimensions is set by specifying an integer greater than zero and less than or equal to the number of rows in the matrix (0 < CD <= N).

The application will automatically detect and correct values greater than the number of rows in the matrix. *Be aware that the more coordinate dimensions generated the longer the algorithm takes to reach completion.* It is highly recommended that 3 dimensions be generated unless there is a reason to generate more.

V M2C 2.1 Output Files

I Definition Files

The definition files created are *TAB delimited* with each input file defined on one line as a series of tab delimited values.

```
timeseries=(true|false)
FlipDistance=(true|false)
InputFile-#=fp   fn   enc-type   delim-type   mx-type  labels  iterations   dimensions
```

The *timeseries*= option specifies if the output is to include the rotation time-series or not.

The *FlipDistance* option specifies the methodology (Flip or Structural Equivalence [SE]) to use when converting strength to distance matrixes.

The *InputFile-#* definition has the following required sub-elements (tab delimited). Any number of files may be specified, however they cannot have the same numerical id.

> # - This is the file numerical id and should be a unique value greater than or equal to zero.
> *fp* – This is the full path and filename for the input file.
> *fn* – This is the filename (only) for the input file.
> *enc-type* – This is the encoding type and can be **ASCII**, **UTF8**, **Unicode** or blank (**other**).
> *delim-type* – This is the type of delimiter used with the file. It can be one of TAB, COMMA, or SPACE (the actual character).
> *mx-type* – This is the matrix type and can be one of **DMatrix**, **SMatrix** or **Coords**.
> *labels* – This is the (true|false) valued flag for labels being present in the input file.
> *iterations* – This is the number of rotation iterations to perform and must be an integer value greater than or equal to one thousand.
> *Dimensions* – This is the number of coordinate dimensions to generate from the raw data and must be an integer value greater than or equal to one.

II Coordinate Files

Coordinate files created will contain a *TAB delimited* list of the coordinates for each label provided with the original matrix. The labels are provided as the first column with each coordinate dimension presented in the columns that follow. Please see the section on Riemann Space for more information about the coordinate system.

III Rotated Coordinate Files

Rotated Coordinate files created will contain a TAB delimited list of the first three dimensions / coordinates for each label provided with the original matrix. The labels are provided as the first column with each coordinate dimension presented in the columns that follow. This option is no longer implemented due to the remove of the rotation option.

VI Bibliography

(9 March, 2009) “Distance Chart USA” 2009, from http://www.mapsofworld.com/usa/distance-chart/.

(4 March 2009). "Rotation (mathematics)." 2009, from http://en.wikipedia.org/wiki/Rotation_(mathematics).

Brandes, U. and Pich, C., Eigensolver Methods for Progressive Multidimensional Scaling of Large Data. Proc. 14th Intl. Symp. * Graph Drawing (GD ’06). LNCS 4372, pp. 42-53. © Springer-Verlag, 2007.

Wasserman, S. and K. Faust (1994). Social Network Analysis: Methods and Applications. Cambridge, Cambridge University Press.

Woelfel, J. and E. L. Fink (1980). The Measurement of Communication Processes: Galileo Theory and Method. New York, Academic Press.