GISpatialNet

GISpatialNet is the brain child of Dr. Eric Jones and Jan Rychtar.
Programmed by Robert Gove, Charles Bevan, and Samuel Tyler.
Collaborators include Martin Smith and Christopher Nicholson.
You can find more information about GISPatialNet at:
http://sourceforge.net/apps/trac/spatialnet/

Table of Contents

General Usage

Command Line Options

GISpatialNet is made to be used either directly from command line options or using a command line interface. The command line options route takes on one of three forms:

```
java -jar GISpatialNet.jar [option]
java -jar GISpatialNet.jar [option] [input file type] [input folder] [output
folder]

java -jar GISPatialNet.jar [input file type] [input folder] [output file type]
[output folder]
```

Where [option] can be:

Option	Function
-c	Starts the command line interface.
-g	Starts the graphical user interface (will be finished in future release).
-s	Performs spatial net bias on files in input folder.
-b	Performs borders analysis on files in input folder (will be finished in future release).
-e	Performs edge highlighting on files in input folder.
-q	Starts the QAP sub program.
-v	Prints the version of GISpatialNet that you are using.
-l	Prints the license information to GISpatialNet.
-h	Prints help for GISpatialNet.

[input/output file type] can be:

Option	File Type
-C	Comma Separated Values (.csv or .txt). This can also handle other forms of separated values (tab, space, etc...).
-D	DL/UCINET (.dat or .txt)
-E	Excel 97' format (.xls)
-K	Google Earth KML format (.kml)
-P	Pajek (.net)
-S	GIS Shapefile (.shp)

For more information on file formats see Input File Formats and Output File Formats. Note that all files in the input folder must be in the exact same format and all the files in the output folder must be in the exact same format.

When running from command line options, input files must have a certain name formatting. The types of files supported from the command line options are files that contain x,y, and attribute data and files that contain adjacency data. X,y, attribute data must end in _xya.ext and adjacency files must end in _adj.ext where .ext is the file type extension. Files that share the same prefix to their filename will be considered part of the same dataset and will be merged before any analysis or output. For example file 1_xya.csv and file 1_adj.csv contain all the data for a data set named “1”. If simple network bias is performed on this dataset, a file named “1.csv” will be produced in the output folder representing the output for all the “1” input files.

Examples

Operation	Command
Start command line interface	java -jar GISpatialNet.jar -c
Read files from “In” folder as CSV files and write to “Out” folder as Google Earth files.	java -jar GISpatialNet.jar -C In -K Out
Read GIS Shapefiles from “In” folder and perform Simple Network Bias analysis on them while writing output to “Out” folder.	Java -jar GISpatialNet.jar -s -S In Out

Command Line Interface

The command line interface is started with the command:

```
java -jar GISpatialNet.jar -c
```

The command line interface is separated into menus that attempt to make using GISpatialNet as easy as possible. The following is a tree representation of some of the menu system.

1. Load Data
    1. Delimited text file (.csv,.txt)
        1. Node data (nodes with attributes)
        2. Adjacency data
        3. Node data only
        4. Attribute data
        5. Main Menu
    2. Excel file (.xls)
        1. Node data (nodes with attributes)
        2. Adjacency data
        3. Node data only
        4. Attribute data
        5. Main Menu
    3. DL/ucinet (.txt,.dat)

    4. Pajek (.net)
    5. Google Earth (.kml)
    6. Shape File (.shp)
    7. Back
2. Save Data
    1. Delimited text file (.csv,.txt)
    2. DL/ucinet (.txt,.dat)
    3. Pajek (.net)
    4. Excel file (.xls)
    5. Google Earth (.kml)
    6. Shape File (.shp)
    7. Back
3. Analyze Data
    1. QAP
    2. Sample Network Bias
    3. Borders
    4. Highlight Edges
        1. Less than average length
        2. Less than median length
        3. More than median length
        4. Top 10%
        5. By value
    5. Matrix Conversion
        1. Translate
        2. Reflect
        3. Rotate
        4. Scale
    6. Back
4. Merge Data
    1. Merge two data sets
    2. Merge all data sets
    3. Back
5. Print Full Status
6. Clear Data

7. About GISpatialNet
8. Add Ego to Data

When working in the CLI each loaded file will be stored in a separate Data Set. The current sets of data are displayed at the top of each menu. Merge data facilitates the ability to load node sets, attribute sets, and adjacency sets separately and then make one Data Set out of the three.

Input File Formats

Comma Separated Values

These files can have either the .csv or .txt extension. If the file has the .csv extension, then the values in the file must be separated by a comma. If the file has a .txt file extension, then the values can be separated by any character (space, tab, /, etc...). You will be prompted on input/output as to what separator is being used. CSV supports all matrix types. CSV files support having multiple matrices of the same type in one file. Each set of matrices must be separated by a line with only one character (usually a /) in it.

The CSV file reader reads a csv file (and text file) using the first row as column labels. It will use any column whose header contains the characters 'ego' or 'Ego' (case sensitive) as the row label. If there is no Ego header, it will use the first column as the row label. The parser will automatically skip the first two fields. The second two fields will be x and y values. All the following fields will be attributes, stored in the attribute matrix.

Here's an example:

ID_Order_of_capture , EgoID , DistX , DistY, Age, Sex
1 , 1 , 3 , 5 , 32 , M
2 , 3 , 2 , 10 , 10 , F
3 , 1 , 3 , 9 , 79 , M

This will be converted in to a XY matrix:

3	5
2	10
3	9

and an attribute matrix:

32	M
10	F
79	M

with the headers being

ID_Order_of_capture   EgoID   DistX   DistY

and the row labels being

1
3
1

Adjacency matrices are distances between the nodes. Only half the matrix is really needed for an undirected graph. For these graphs, the diagonal is either zeros or ones. These files are split by Ego ID, being they make a new Matrix for each Ego, so the number of egos in the file should match the number of columns plus the id and ego_id. The final Matrices should be: [number of egos] matrices, each with [number of egos]x[number of egos] matrices. (i.e. if you have 5 egos, you should have 7 headers, and each)They should be in the form:

```
ID_Order_of_capture , EgoID , 1 , 2 , 3
1 , 1 , 1 ,
2 , 1 , 1 , 1 , 10 , F
3 , 2 , 3 , 2 , 1 ,
4 , 2 , 2 , 2 , 10 , 1 , F
5 , 3 , 3 , 3 , , , ,
6 , 3 , 2 , 3 , 10 , F
```

DL/UCINET

These files closely follow the DL/UCINET standard. All DL/UCINET files have a .dat or .txt extension. From the DL/UCINET standard, block-matrix format, data file references, and matrix labels are not supported. When using a DL/UCINET file, make sure that all headers are of the form “header= x” or “header = x”. In other words there must be a space between the “=” and a value of an item in the header. The DL/UCINET format only supports adjacency matrix types. DL/UCINET supports having multiple matrices of the same type in one file. Each set of matrices must be separated by a blank line.

Excel 97'

These files follow the Microsoft Excel 97' format exactly. Excel files have a .xls extension and support all matrix types (see Matrix Types, page 9). Excel files are similar to CSV files and support having multiple matrices of the same type in one file. Each set of matrices must be on a separate worksheet.

Google Earth

These files follow the Google Earth KML standard format approved by GIS. Google Earth files have a .kml extension (Note: some Google Earth files have a .kmz extension. While not supported by this program, a .kmz file is simply a zip file of .kml files). Only a subset of the full KML format is supported:

Supported Tags:

- Point
- name
- description
- scale

- Placemark
- text
- displayName
- href
- linkDescription
- linkName
- message
- near
- phoneNumber
- value
- viewFormat
- color
- hotSpot

A KML file will fill a x matrix, y matrix and attribute matrix when given the correct data. For example the point, hotSpot, and placemark tag will fill both the x and y matrix while every other tag will fill and attribute.

Pajek

Pajek files have the extension .net and fill all matrices except an attribute matrix. For a Pajek file, GISpatialNet supports only node lists and edge matrices. No extra information attached to nodes is supported.

Example:

```
*Verticies
1 0
0 1
1 1
0 0
*Arcs
1 0 0 0
0 1 1 1
0 1 1 0
0 1 0 1
```

The Arcs matrix must be a full matrix.

GIS Shapefile

GISpatialNet uses the GeoTools library to read and write Shapefiles. Shapefiles have a .shp extension and come with many other files like a .shx or .db. The other files hold extra information such as data or project information used in GIS ArcMap. GISpatialNet allows point, multipoint, line, and multiline geometries to be used in a shapefile. Depending on the data provided, a shapefile can fill any matrix. A point/multipoint geometry will fill the x and y matrix while a line/multiline will fill the adjacency matrix. All extra data in the database file will go to the attribute matrix.

Output File Formats

Comma Separated Values

Depending on options that are chosen, a csv or .txt file will be written containing data from either the x,y, and attribute matrices or the adjacency matrix. Each data set in memory will be written to a separate file, writing multiple data sets to a single file is not supported.

DL/UCINET

Directly writes a .dat file that can be read in DL/UCINET. Each data set in memory will be written to a separate file, writing multiple data sets to a single file is not supported.

Excel 97'

Depending on options that are chosen, a .xls file will be written containing data from either the x,y, and attribute matrices or the adjacency matrix. Each data set in memory will be written to a separate file, writing multiple data sets to a single file is not supported.

Google Earth

Directly writes a .kml file that can be imported to Goole Earth. Writing a Google Earth file Requires that the attribute matrix contains a Name and description as the first two attributes. Writing of edges is not supported. Each data set in memory will be written to a separate file, writing multiple data sets to a single file is not supported.

Pajek

Directly writes a .net file that cen be imported to Pajek. Each data set in memory will be written to a separate file, writing multiple data sets to a single file is not supported.

GIS Shapefile

Known Coordinates

Writes two shapefiles that can be imported to GIS ArcMap or other programs. The first shapefile contains all the nodes with attribute data and the second contains all the edges.

Unknown Coordinates

Generates X and Y coordinates based on an algorithm chosen by the user and then writes two shape files that can be imported to GIS ArcMap or other programs. The first shapefile contains all the nodes with attribute data and the second contains all the edges.

Algorithm Information

Border

This is a port of the original border program by Franz Manni PhD. Barriers can be found at http://www.ecoanthropologie.cnrs.fr/spip.php?article91.

Quadratic Assignment Procedure

The is a port of the original program implementation of the Quadratic Assignment Procedure (QAP). The algorithm was proposed by D. Krackhardt in the article “Predicting with networks: Nonparametric multiple regression analysis of dyadic data (1988)”. Running QAP requires files in the same format as is required by the original ZT Win program.

Edge Highlighting

This function allows the user to create two shapefiles, one containing edges that meet a certain requirement and one that contains edges that do not meet a certain requirement. The requirements are preprogrammed as:

1. Less than average length
2. Less than median length
3. More than median length
4. Top 10%
5. By value

If Highlighting by value is chosen, a shapefile for each distinct value in the adjacency matrix is created (including a shapefile that contains edges that represent the value 0).

Simple Network Bias

This is an implementation of an algorithm proposed by Eric Jones Ph.D. and Jan Rychtar Ph.D. This algorithm was originally implemented by Robert Gove and later imported into GISpatialNet.

Matrix Types

X and Y matrix

The X and the Y matrix are both vector matrices that hold only coordinate data. Since the data in the X and Y matrix is not checked, it is possible to enter either actual (x,y) space data, UTF coordinate data, or any other form of data where space is marked with two numbers.

Adjacency Matrix

The adjacency matrix gives values to edges between nodes in the x and y matrices. The adjacency matrix can take on one of three forms; Full, Upper, and Lower.

For example if we have three nodes: (0,0), (1,1), and (2,2) where there is a link between a node and itself and there is a link between (0,0) and (1,1).

Full Matrix:

1 1 0

1 1 0

0 0 1

Upper Matrix:

1 1 0

  1 0

    1

Lower Matrix:

1

1 1

0 0 1

The numbers in the adjacency matrix can be any whole number and usually represent the intensity of the link.

Attribute Matrix

The attribute matrix can be a set of any number of attributes where the row of the attribute matrix corresponds to the matching rows in the x and y matrix that make up the coordinates of the node for which the attributes belong to. Attributes can be anything (numbers, words, etc...).

Troubleshooting

General

If GISpatialNet seems to be running slow and you know that you have a good computer, you can increase the amount of memory allocated by adding the option -Xms <size> before the -jar option on the command line:

```
Java -Xms 1024m -jar GISpatialNet.jar ...
```

The above line of code sets the memory allocated to the program to 1024 Mb (or 1Gb). Sometimes when working with large sets of data it is important to increase the amount of memory to reduce the number of times that data is written and read from the hard drive (subsequently increasing speed).

Bug Reporting

To file a bug go to http://sourceforge.net/projects/spatialnet/develop and click on the File a bug link on the right hand side of the page under “Links You May Need” column.

Dependencies

- UJMP -- Universal Java Matrix Package
- GeoTools -- Geographical Tools
- Mindprod CSV Writer -- Excellent CSV Writer
- KML framework -- Google KML in java project
- JExcel Api -- Excel reader/writer written in java
- GetOpt -- CLI option helper tool

License Information

This software is governed under the GNU Greater Public License, Version 2 (GPLv2). If you have not obtained the LGPL with this software, you can obtain it from http://www.gnu.org/licenses/gpl.html. GISpatialNet uses software governed under the GPL, LGPL, Apache License, New BSD License

Where To Get Extra Information

- http://www.gnu.org/licenses/gpl.html
- https://sourceforge.net/projects/spatialnet/
- https://sourceforge.net/apps/trac/spatialnet/wiki