

Libraries Used

Python 3.6+ was used in development of this plugin. The following are the python libraries used:

```
networkx==2.4  
pysal==2.0.0  
PyQt4==4.11.4  
PyQt5==5.15.1
```

GitHub Repository for StreetAnalysis

The GitHub location of StreetAnalysis can be found here:
<https://github.com/maltaweel/StreetAnalysis>

Intent of Models

The models applied provide a relatively rapid and simple way to assess street networks to see where segments are likely to have greater traffic (pedestrian) based on relative distance within urban contexts. The intent is to demonstrate where traffic may imply social interaction or reflect social significance based on space syntax methods. Additionally, algorithms apply different measures of centrality and efficiency in measuring the street network using well known graph theoretical methods. The intent is to produce measures that are comparable and that can be statistically assessed (e.g., distribution of nodes travelled, centrality variations, etc.).

Data Requirements

To run the modules, you need to have a shapefile that has snapped street segments where each link is two nodes consisting of a start and end node. This will make the street segments a graph that is analysed in the modules. The outputs of the models are .csv files that have the point data (the nodes) of the street segments that are then associated with a given analysis (e.g., number of times visited). You can vectorize the outputs to QGIS or other GIS platforms if you like or simply do statistical analyses of the data.

Key Analysis Code:

Street Network Analysis

The first analysis is the Street Network Analysis, which applies the loadApplyModel.py module. This is an iterative model whereby every node is used once as a starting node and each nodes is also a destination. The intent is to determine which nodes, and thereby segments, are traversed the most. The nearest path analysis uses a Dijkstra algorithm. This method is comparable to the Standard Decision Model used in Altaweel & Wu (2010), except no metabolism and agent speed is used. Additionally, the assumption is that the surface is relatively level or elevation is of minor consequence. Thus, distance is how edge weights, which are street segments, are determined. After the model is completed, an output file (called results.csv) is produced which has the nodes (x and y values) and number of times the node was traversed.

Road Graph Analysis

In this analysis, a graph is created which is used to study the street network. This is applied in the networkAnalysis.py module. The different analyses applied are: global network efficiency, centrality measures, which includes betweenness (edges and nodes), closeness, and degree

centrality, efficiency centrality, and straightness centrality. All of these are outputs for nodes, while betweenness does both nodes and edges. The results are provided as .csv files (globalEfficiency, efficiencyCentrality, straightnessCentrality, edgeBetweennessCentrality, and nodeCentrality). The algorithm applies the methods as discussed in Port et al. (2006). This provides another set of measures to compare between different networks or graphs. For further detail, see networkAnalysis.py.

Other Folders

HTML_Documentation: This provides the html pydocs which provide information on the methods applied in relevant modules.

Sample_Data: Sample street network data to test the analyses.

References: This provides the references used to shape the algorithms applied and discussed above.

References

Altaweel, M; Wu, Y. (2010). Route selection and pedestrian traffic: applying an integrated modeling approach to understanding movement. *Structure and Dynamics: eJournal of Anthropological and Related Sciences* 4(2). Retrieved from <https://escholarship.org/uc/item/6898p5vm>.

Porta, S.; Pablo, C.; Vito, L. 2006. The Network Analysis of Urban Streets: A Primal Approach. *Environment and Planning B: Planning and Design* 33 (5): 705–25. <https://doi.org/10.1068/b32045>.