Visualizing Social Networks with Hatchet

Mohamed Aamir
maamir@umd.edu

Johann Antisseril
johann@umd.edu

Jasmine Parekh
jparekh2@umd.edu

Omer Sharif
omer2000@umd.edu

Shelton Zhou
sz2001@umd.edu

ABSTRACT

UPDATED—December 16, 2022. Social networking rates and social media figures are still on a global, upward trend. As of January 2021, the world's most popular social networking sites - Facebook, Youtube, and Whatsapp all report over 2 billion active users [10]. And as of February 2021, 72% of U.S. adults have reported using at least one social media site daily [8]. Social media makes it possible to connect and interact with people around the world, family, friends, and strangers alike. The possibilities for social media content are endless as one can share opinions, agreements, disagreements, pictures, personal details, endorsements, and as the last few years most popularly, news [6]. As social media popularity continues to increase, the web of connections between users only becomes more complex and tangled, making social media visualization that much more important but also harder and inefficient. In this paper, we aim to test if the python Hatchet library [1], which originally analyzes hierarchical performance data, can be given a new use case in terms of visualizing social networking data. First, we will present research on other social media visualization tools and highlight their fundamental functionality. Our new tool must perform at least at this level to be considered competitive. Second, we attempt to sum all relevant information and documentation on Hatchet in order to familiarize ourselves with the technology and apply it to sample data sets. Lastly, we will draw conclusions that as of right now Hatchet is not capable of producing the graphs needed to visualize social media networks.

1. INTRODUCTION

To truly understand a social media network is absolutely critical to visualize it. Otherwise, the raw data is just rows and rows of words and numbers that will not reveal any sort of insights no matter how many hours are spent combing through it. Facebook, for example, generates 4+ petabytes, one million gigabytes, of data per day. To put that into perspective, the most any iPhone can hold is around 500 gigabytes so, Facebook gets 2,000 iPhones worth of data on any given day [2]. Therefore these insights, in the form of trends, patterns, or outliers, can help companies make incredibly important decisions on topics such as marketing, technological improvements, new feature needs, campaigns, and general feedback on day-to-day operations. Not only is this type of visualization needed by social media companies themselves, but virtually any mid to large-size company could benefit from analysis of their online presence throughout social media. Furthermore, while the main applications of this tool are directed at social media, it does not necessarily have to be limited to it. A social media network when represented visually boils down to a graph data structure, a web of nodes connected by edges. There are many other uses for graph visualization such as business networks, product distribution, or recommendation engine algorithms.

The problem with the standard for visualization today is that actually, this "standard" doesn't exist. There is no one way or even best way to represent a social network. A "better" method is currently defined as a method that is more time-efficient, more space-efficient, easier to replicate on a range of data, and easier to understand post-visualization. If a new solution even achieves one of these aspects then it could raise the bar for the next developed tool, but to become the standard the tool perhaps requires all 4 aspects. The road to innovation in this space is by experimenting with new techniques and technology and then rigorously testing its competitiveness with existing solutions.

In this case, we will use a relatively new technology, Hatchet, that primarily focuses on the analysis of hierarchical data. In terms of visualization, Hatchet works with pandas dataframe to store the data and can convert hierarchical raw data into various visual charts such as a tree, directed graphs, and flamegraphs. The extra incentive of Hatchet is any visualization can become interactive with the addition of one line of code making it possible to hover over nodes to get extra information and get a 3-D visualization. All of this sounds great on paper, however there are significant challenges faced when attempting to use new technology. For one, documentation is scarce and incomplete, which in turn makes troubleshooting harder. Second, many of the features described such as the interactive feature or another filter feature are documented as "still in the research stage" and currently undergoing development which can hinder the tool's application to this use case [18]. Lastly, and most importantly, Hatchet deals in hierarchy which dictates order and direction within its visualized data structure. However, social media data is not often presented this way. It may take significant effort to convert data into this fashion or devise a new way that the current data can fit into the Hatchet mold.

In order to tackle these challenges we aim to spend significant effort on research to be able to brainstorm ideas to circumvent the issue at hand. One working theory is that if Hatchet only produces directed graphs then could social media's undi-

rected network be represented as a directed graph in Hatchet only with directed edges in both directions, outgoing and incoming, between every pair of connected nodes. This possible solution brings up a new level of issues such as how would this affect storage and runtime as well as if such degree of effort is worth it in comparison to other competitors. In this fashion, we plan to introduce potential solutions, only to research them to develop a pros and cons list for each. Then we will put the top methods to test with code and troubleshooting, and evaluate the results.

2. RELATED WORK

In this section, we present the summary of multiple related academic papers to the visualization of social networks. As well as, analysis of some visualization tools currently available for public use that present as learning opportunities for the creation of a Hatchet tool and as potential competition for it.

2.1 Graph Visualizations

The purpose of a graph or any other method of visualization is to help organize the data in an effective way. In the paper *Interactive visualization of large graphs and networks* [12], effective visualizations of large data should follow the following rules: it should have a clear structure, background context on the organization of the graph, and the nodes of a graph should be connected in some way. When a graph has a clear structure, it allows the user and others to visualize the data properly. It can lead to a better understanding of random data and can possibly lead to making connections and thus a relationship.

The purpose of having a clear background context is to help the reader understand how exactly the data is being organized and what each thing on the graph means. For example if there is a graph where one node is depicted larger than another node, there should be enough information given to the reader to let them know why this is the case. Another example is if an edge between two vertices is more bold or thicker than other edges, it should be either implied or given that this could mean that there is a significant meaning to this occurrence in the graph. The final point is that the nodes should all be connected in a relevant way. Whether it is as simple as each node is connected to each other or if nodes are clustered together and represented as connected, the graph should have a meaning when nodes are connected to either show a relationship or proximity to each other in nature. These elements combined lead to an effective visualization of any sort of data.

2.2 Social Network Analysis with Graph Applications

Social Network Analysis (SNA) [14], is mostly studied through big data analytics with content-based analysis and text mining from textual data. Content-based analysis focused on extracting intelligence from the content formed and collaborates in the network. There are two main types: Audio or speech analysis and video content analysis that first extract information from unstructured audio data sets while the other involves techniques to monitor, analyze, and extract information from video streams. Image analysis does face recognition and sentiment extraction from social media data.

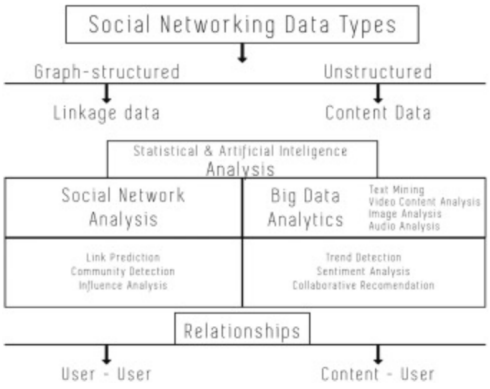

Figure 1. The graph shows the difference between SNA and Big Data Analysis (BDA). SNA is analyzing relationship between users,whereas BDA is analyzing between the user and their contents to form a summary. Image Source: [14]

Text mining extracts data from textual data which is often incorporated with the analysis above. One that Online Social Network (OSN) analysis uses the most is image analysis and text mining since users often post images with captions as the keywords to the image. So it will get sorted and the same will happen with video analysis with text mining. Content data in different parts of a network is often related to its structure, so combining both sources of content mining and SNA is considered to perform better in analysis.

SNA, is a term that encloses structure-based analysis and is similar to structural analysis. The paper is to help us configure whether it will help one to understand the structure of the network to gain insights about how it works and make decisions upon it. As a result, nodes/link characteristics or metrics forms by looking at the entire network cohesion will be the best format for SNA. Comparing and tracking changes in the network over time, revealing communities and key nodes, and determining positions between individuals and crowds are its common procedures and outcome. These involve static or dynamic analysis. A social network can change gradually over time and analysis on the entire network can be done in "batch mode". Conversely, dynamic analysis, which is more complex, includes streaming data that are evolving in time at a high rate. Dynamic analysis is often in the area of interactions between entities whereas static analysis deals with properties like connectivity, density, degree, diameter and geodesic distance.

Mining the content of OSN in conjunction with the network can be useful in efficiently answering sub questions of an analysis such as "Who has the most influences (connections) on a social media platform?" or "Is there a trend or relationship between the similar contents from users?". Graph based mining tools are required in order to easily model the structure of the social networks and perform the above tasks. There is a great variety of software tools that analyze properties of nodes and edges in a network- NodeXL, NetworKit, Pajek, and Statnet. Some of the tools were originally devel-

Figure 2. Each one of the three graphs represent possible graphical interpretations of the 3D Hyperbolic Quasi-Hierarchical Graphs. As shown, each representation can clearly show the relationship between parent and child nodes in 3D space. Image Source: [12]

oped for network visualization, and now contain analysis procedures and others were specifically developed to integrate network analysis and visualization.

2.3 Graphical Representation of Social Networks

Social networks are highly connected in nature as everyone these days knows so many other people. Since there is so much inter-connectivity there needs to be a way to create a way to visualize this data. In the paper, Interactive visualization of large graphs and networks, the ideal way of representing any large set of data optimally would be by using 3D hyperbolic quasi-hierarchical graphs. A 3D graph helps give the view an extra dimension to help create a better spatial analysis of the graph being produced. It would help reduce the number of collisions between nodes as there is significantly more space to represent nodes with a better way to position each node in a more meaningful way. A hyperbolic representation gives an elegant way to represent the nodes and gives an exponential space for points to plot where the distance between each node is the same. Hyperbolic geometry allows for multiple parallel lines that run through a point which can be useful for high volumes of data. The quasi-hierarchical representation mentioned in this paper was a tree. This graph gives a relationship between nodes where each node after the root has a parent and at least one child until it reaches a leaf node. By creating a 3D hyperbolic tree we can achieve handling a lot of data where multiple connections can be made. It will ensure that each part of the tree is readable and makes sense. There will be enough space to show multiple nodes and make it clear for people to valid connections. It would theoretically look as shown in Figure 2.

2.4 Different Graph Analysis Tools

There are quite a few graph analysis tools out there. Some of them include Pajek [4], Cytoscape [15], Osprey [7], CNPlot [3], Medusa [9] and Depthmap [16]. All of these tools have their advantages and disadvantages. Let's try to dig deeper into some of these tools.

Medusa [9] was an interactive graph analysis tool that was created to facilitate different types of graph analysis in biological studies. Some biological processes it was able to analyze when viewed as graphs were metabolic pathways, gene regulation, or protein-protein interactions. The main problem

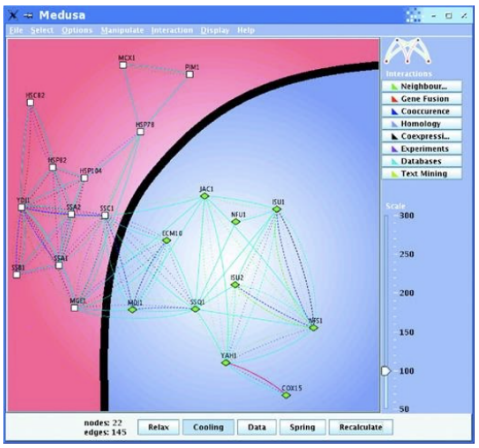

Figure 3. Investigating the evidential and physical links between proteins transversing the mitochondrial-cytoplasmic boundary in Yeast. Details from work in progress (Lars Steinmetz, personal communication). For instance, ECM10 (a homolog to heat shock protein 70) is linked by gene neighbourhood, co-expression and text mining to the mitochondrial precursor protein MDJ1

Medusa [13] was trying to solve was reducing the gap between visualization and network analysis. The research found that it was a powerful tool for visualization and clustering analysis of large-scale biological networks. It best works with data from the STRING database but it is said that it can be used for almost any kind of graph. The STRING database is a database that contains known and predicted protein-protein interactions. Medusa was written in Java. It is available both as a standalone application and as a Java applet. Medusa was designed to be simple, which is why it did not contain a wide range of functionality. It was supposed to be an easy way to visualize graphs in a web setting. However, it still supports different types of graph settings like directed, weighted, multi-edged, undirected, unweighted or certain combinations of the above. It also helps in clustering analysis of biological networks that are great in size. It is supposed to be very user-friendly. It is open source, which gives a lot of programmers the opportunity to change and extend its functionality.

Depthmap [17] is another tool in the graph analysis paradigm. It was created for the purpose of performing visibility graph analysis of spatial environments. It was made in 2001 and has undergone some developments ever since. When it started off, it was first developed for Windows 95/98/NT and 2000 platforms. It allowed the user to import layouts in 2D DXF(drawing exchange format), add points or nodes to the layout in Depthmap, and perform graph analysis.

When Depthmap was first introduced as a graph analysis tool, it provided a few ways to "measure" graphs. Some of these measurements included clustering coefficient, control, mean depth, and point depth entropy. Clustering coefficient of a vertex 'v' was defined as the number of vertices that were connected within the neighborhood of 'v' divided by the highest possible number of edges within the neighborhood of 'v'. For example, if the neighborhood of 'v' con-

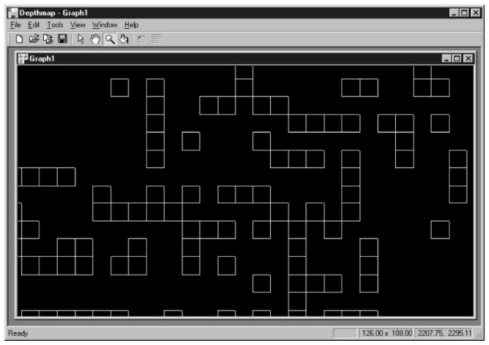

Figure 4. The application as it appears after the DXF file has been imported

tains 10 vertices and there are 30 edges between vertices in the neighborhood of 'v'. The highest possible number of vertices in this neighborhood, if every vertex was connected to every other vertex, would be 10*9 = 90. The clustering coefficient would be 30/90 = 0.33 according to the definition. Control for a location was defined as the sum of the reciprocals of neighborhood sizes of the vertices that were connected to the current vertex. For example, if a vertex 'v' had 3 neighbors whose neighborhood sizes were 2, 4, and 5 respectively, its control would be 1/2 + 1/4 + 1/5 = 0.95 according to the definition. The mean depth of a vertex is the average of the number of edges from the current vertex to every other vertex. Note that only shortest paths are considered in calculating the mean depth. The point depth entropy was used as a way to measure the order around a particular point in the graph.

The creators of Depthmap [16] decided to include the above measurements because they believed that these ways to measure graphs had applications in graph-related projects. In 2007, some additional functionalities were added to UCL Depthmap 7, which was intended to be used for isovist analysis, to make it suitable for generic spatial network analysis. Some of these features include axial retrieval and analysis, segment analysis, data layers, scattergrams etc.

3. SOLUTION

3.1 The Dataset

The dataset we chose for this project is taken from SNAP [11] and is based on a network of anonymous people who trade using Bitcoin as the medium on a platform named Bitcoin Alpha. The users in this dataset are anonymous and instead of storing users' information, this dataset maintains a rating-based metric on a scale of -10 to 10 to keep track of transaction quality and monitor fraudulent activity. There are about 3,783 nodes in this graph along with 24,186 weighted edges. These edges have 93% of positive edges that are between the edge weight ranges -10 and 10. Each row in the dataset has four elements - ***source***: node id of source (rater), ***target***: node id of target (ratee), ***rating***: the source's rating for the target ranging from -10 to +10 and ***time***: the time of the rating.

	source_id	target_id	rating	time
0	1	1753	-10	1.333403e+09
1	1	1771	-10	1.339444e+09
2	1	2096	-10	1.341116e+09
3	1	2410	-10	1.345780e+09
4	1	2471	-10	1.345780e+09
5	1	1383	-10	1.411966e+09
6	1	672	-5	1.306513e+09
7	1	62	-5	1.411966e+09
8	1	905	-5	1.411967e+09
9	1	15	1	1.289243e+09
10	1	32	1	1.290667e+09
11	1	34	1	1.290750e+09
12	1	54	1	1.292196e+09
13	1	71	1	1.293481e+09
14	1	74	1	1.293579e+09
15	1	78	1	1.293750e+09
16	1	81	1	1.294982e+09
17	1	101	1	1.295164e+09
18	1	119	1	1.296512e+09
19	1	138	1	1.297374e+09
20	1	110	1	1.297627e+09

Figure 5. Dataframe generated from edge lists

3.2 Our Approach

This is a social network based on a user-to-user network based on trust/distrust of bitcoin trading. The trust/distrust metric is measured based on edge weights where positive edge weights indicate that the source is trustworthy and negative edge weights denote distrust.

Hatchet's primary data structure is a *GraphFrame* object which is a combination of a graph with a pandas *DataFrame* [5]. Our main goal is to figure out fraudulent activities in the Bitcoin Alpha platform by analyzing the chosen dataset using Hatchet's DataFrame operations like filter and GraphFrame operations like squash for understanding the characteristics of the social network along with the help of the graph implementation of Hatchet [5]. Our initial approach was to create a DataFrame from the edge lists given to us by the dataset and try to convert it into a GraphFrame for analysis. The aim is to find ways to incorporate current social network graphs in the Hatchet library by modifying certain operations and methods in Hatchet.

This social network dataset is smaller than most social networks because integrating the dataset into a GraphFrame object is quite challenging. This is because Hatchet currently accommodates directories like the HPCToolkit databases, DOT files, and Caliper Cali files which are generated to scale performance rather than mapping social network graphs.

4. RESULTS

We were able to create the DataFrame for the edge lists (Figure-5), but converting the DataFrame into a GraphFrame object was challenging because we had to modify the functionality of GraphFrame in Hatchet to give us a graph visualization.

From the graph output (Figure 6.), the colored values are the trust/distrust ratings, where red indicates distrust *rating* and

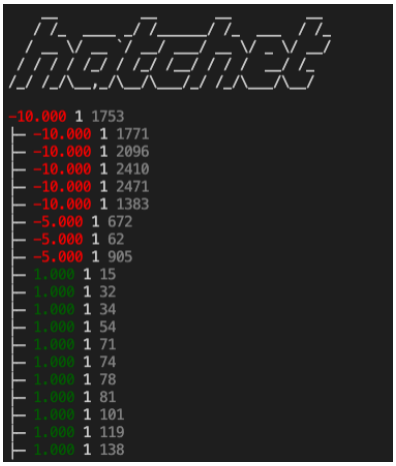

Figure 6. Modified version of Hatchet's graph visualization

green indicates trust *rating*. These ratings are followed by *source* and *target* node values. The DataFrame used to derive this graphical visualization contains 35592 rows of edges and we can notice that the output graph is not useful for a good analysis because of the current implementation of the graph visualization. Upon integrating features that support visualizations for social media graphs, Hatchet may support a more useful analysis for social networks.

A problem we encountered using Hatchet is that it stores the graph data in hierarchical order and this makes it difficult to parse directed graphs into a GraphFrame data structure. We intended to take the dataset and modify the structure to fit the GraphFrame object for Hatchet to generate a graph for analysis. The key problem being run into is attaining the graph output for the social media graph. Since Hatchet is a relatively new library with little documentation, getting Hatchet running for this project was difficult. Manipulating data using Hatchet and displaying it turned out to be a huge roadblock while completing this and due to this, it prevented the intended plan for the project from being carried out. After trying different ways to generate a GraphFrame object, we were able to convert the edge list DataFrame to a graph visualization (Figure 6.).

Based on our understanding of how Hatchet is functioning, we believe that adding a feature to analyze social media graphs does not support the datasets that are currently available. The graph visualization of Hatchet is designed in a hierarchical form forcing the social network graph to look like a long and narrow graph (Figure 6.). This makes it difficult to analyze the graph output since it did not truncate information for better analysis as we expected. Hatchet is still evolving, so there is room to integrate features that make a graph visualization that supports social media network analysis.

The data used in most of the examples using Hatchet have information regarding factors such as node names and children names. But most datasets correlated to social media network graphs do not have such elements as they only consist of edge lists and their weights most of the time. This allows Hatchet room for incorporating options to include graph visualization by allowing users to specify values to be shown in the output graph.

5. CONCLUSION

In this project, there was a struggle to understand how Hatchet worked, it was generally very difficult to get the code working the intended way for this project. Some future steps that can be taken for this project is to better understand how the GraphFrame data structure works and manipulate it to work it in the way necessary to get the output. Another possible step is to wait and let the Hatchet library mature and let it develop more. The biggest issue the group ran into while working on this project was that Hatchet was so new with barely any documentation, it was hard to make the progress needed and know what to do when stuck. It is a possibility that Hatchet will be expanded to be more flexible with accepting more types of inputs. However, while working on this project, it is clear that Hatchet can possibly be a good graphical visualization, but with the given resources, it is tough to see if Hatchet can support graphing a social network effectively. Since Hatchet is so rigid, relatively new and hard to use, there is a possibility that other options need to be explored as to how to best create a visualization of a social network data. In the future, with whatever chosen method, there will be a point where the visualization will need to be tested to see if it is effective. To do that, a survey will be conducted among random people where participants will be asked the question "Which graph best represents a social network. Choose one which clearly shows relationships and where you can effectively draw conclusions from." They will be given multiple different types of graphs and be asked to choose one which they think is best. If a majority of the participants choose the graphical representation created by the group, it would be considered a success such that the graphical method chosen by the group is likely an effective visualization of social network data. The purpose of this survey is to ask random people from various backgrounds to see if they can make sense from a visualization as the purpose of a visualization is to help the viewer to understand the data and to draw conclusions from it. If random people from various backgrounds are able to do that, this is a success. If the chosen graphical method is not chosen, then it can be concluded that it was not an effective visualization.

REFERENCES

[1] 2021. Hatchet. (2021). https://hatchet.readthedocs.io/en/latest/

[2] 2021. Wild and Interesting Facebook Statistics and Facts (2021). (Jan 2021). https://kinsta.com/blog/facebook-statistics/

[3] N. N. Batada. 2004. CNplot: visualizing pre-clustered networks. *Bioinformatics* 20, 9 (2004), 1455–1456. DOI:http://dx.doi.org/10.1093/bioinformatics/bth080

[4] Vladimir Batagelj and Andrej Mrvar. 1998. Pajek-program for large Network analysis. *Connect* 21 (01 1998), 47–57.

[5] Abhinav Bhatele, Stephanie Brink, and Todd Gamblin. 2019. Hatchet: Pruning the overgrowth in parallel profiles. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. 1–21.

[6] Lauren Bratslavsky, Nathan Carpenter, and Joseph Zompetti. 2019. Twitter, incivility, and presidential communication: A theoretical incursion into spectacle and power. *Cultural Studies* 34, 4 (2019), 593–624. DOI:http://dx.doi.org/10.1080/09502386.2019.1656760

[7] Bobby-Joe Breitkreutz, Chris Stark, and Mike Tyers. 2003. Osprey: A Network Visualization System. *Genome biology* 4 (02 2003), R22. DOI: http://dx.doi.org/10.1186/gb-2002-3-12-preprint0012

[8] Pew Research Center. 2021. *Demographics of Social Media Users and Adoption in the United States*. (23 November 2021). https://www.pewresearch.org/internet/fact-sheet/social-media/ Accessed: 2018-12-09.

[9] S. D. Hooper and P. Bork. 2005. Medusa: a simple tool for interaction graph analysis. *Bioinformatics* 21, 24 (2005), 4432–4433. DOI: http://dx.doi.org/10.1093/bioinformatics/bti696

[10] Simon Kemp. 2021. Digital 2021: Global Overview Report - DataReportal – Global Digital Insights. (Oct 2021). https://datareportal.com/reports/digital-2021-global-overview-report

[11] Jure Leskovec and Andrej Krevl. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. http://snap.stanford.edu/data. (June 2014).

[12] Tamara M. Munzner. 2000. *Interactive visualization of large graphs and networks*. Ph.D. Dissertation. https://www.proquest.com/dissertations-theses/interactive-visualization-large-graphs-networks/docview/304651820/se-2?accountid=14696 Copyright - Database copyright ProQuest LLC; ProQuest does not claim copyright in the individual underlying works; Last updated - 2021-09-28.

[13] Georgios A Pavlopoulos, Sean D Hooper, Alejandro Sifrim, Reinhard Schneider, and Aerts. 2011. Medusa: A tool for exploring and clustering biological networks. *BMC Research Notes* 4, 1 (2011). DOI: http://dx.doi.org/10.1186/1756-0500-4-384

[14] Anni Sapountzi and Kostas Psannis. 2016. Social Networking Data Analysis Tools Challenges. *Future Generation Computer Systems* 86 (10 2016). DOI: http://dx.doi.org/10.1016/j.future.2016.10.019

[15] Paul Shannon, Andrew Markiel, Owen Ozier, Nitin S. Baliga, Jonathan T. Wang, Daniel Ramage, Nada Amin, Benno Schwikowski, and Trey Ideker. 2003. Cytoscape: A Software Environment for Integrated Models of Biomolecular Interaction Networks. *Genome Research* 13, 11 (Nov 2003), 2498–2504. DOI: http://dx.doi.org/10.1101/gr.1239303

[16] Alasdair Turner. 2001. Depthmap: a program to perform visibility graph analysis. In *Proceedings of the 3rd International Symposium on Space Syntax*, Vol. 31. Citeseer, 31–12.

[17] Alasdair Turner. 2007. UCL Depthmap 7: From isovist analysis to generic spatial network analysis. *New Developments in Space Syntax Software, Istanbul* (2007).

[18] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. 2005. Learning from labeled and unlabeled data on a directed graph. *Proceedings of the 22nd international conference on Machine learning - ICML 05* (2005). DOI: http://dx.doi.org/10.1145/1102351.1102482