Porto Business School
/ University of Porto

Social Network Analysis

Part 2 – Principles of Network Analysis and Visualization

Prof. Eduarda Mendes Rodrigues

Porto Business School
/ University of Porto

Part 2 - Principles of Network Analysis and Visualization

- Challenges in analyzing social network data
  - Data collection, representation, scale considerations
- Process model for social network analysis
- Overview of network analysis toolkits
- Hands-on tutorial

Challenges in Analyzing Social Network Data

Network Analysis Challenges

Porto Business School / University of Porto

Data comes in many different formats, shapes, speeds

- Implicit relations, explicit relations, multi-modality, multi-relational
- Nodes could be people, text, documents, organizations, websites, blogs, etc.
- Manual (self-report, observation), data sets, automatic network extraction (NLP) – **validation!**

Network Analysis Challenges

- **Which network representation can help explain social phenomena at a large scale?**
  - Information diffusion, infectious diseases, polarity of opinions in politics, disaster mitigation, etc.
  - Use benchmark data sets vs. extract own data
- **Size and complexity!**
  - Social media networks can comprise millions of nodes
  - Networks evolve over time (more nodes, link formation)
- **Privacy concerns, ethics**

Network Analysis Challenges

- **Data formats interoperability, data integration, data management**
  - different tools use different data formats, conversion may be required
- **Computational requirements**
  - some tools are not able to cope with more than a few thousand nodes
  - big data processing challenges (volume, velocity, variety)

Network Visualization Challenges

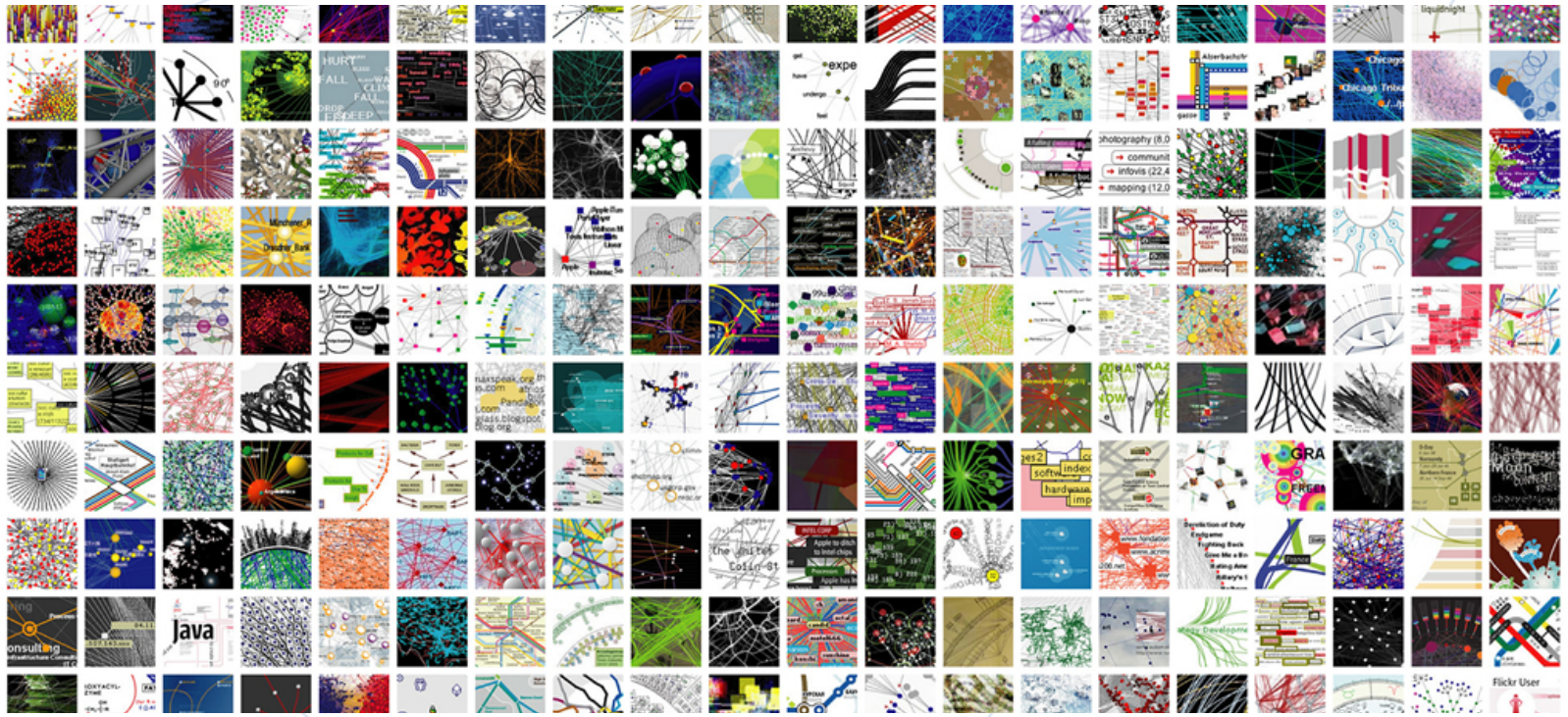
- **Ideally…**
  - Every node is visible
  - The degree of every node can be counted
  - It is possible to follow every link from source to destination
  - Clusters and outliers are identifiable

NetViz Nirvana!

Dunne and Shneiderman (2009)

Network Visualization Challenges

- Real networks are typically very complex structures!



How good is a network visualization?

- Edges crossings and node occlusions
  - standard layout algorithms don't help much when the size of the network is above a few hundred nodes and the network is relatively dense in the number of links
- Lack of contextual information
  - interpretation of the network structure often requires visualizing additional information about the nodes and links
- Network layout algorithm
  - an inadequate choice of network layout algorithm may hinder the discovery of patterns in the network structure

How good is a network visualization?

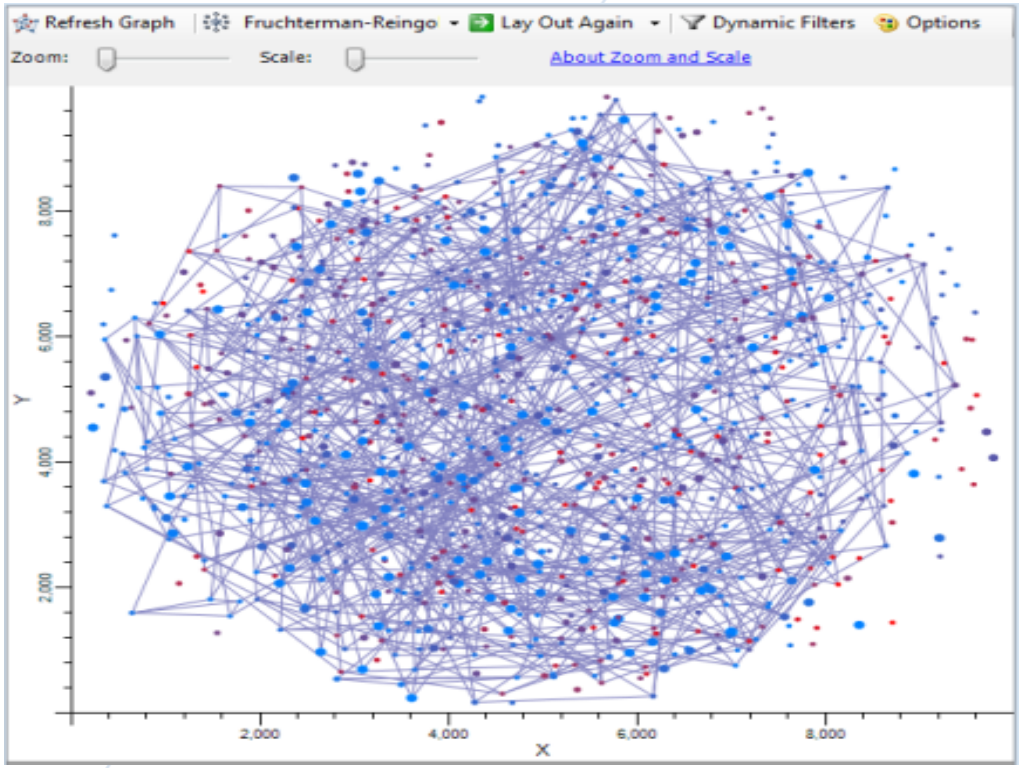
- E.g. Sierpinsky Triangle

  - **A fractal named after W. Sierpinski who described it in 1915**

  - Self-similar pattern, replicated at every level of reduction or magnification

How good is a network visualization?

- E.g. Sierpinsky Triangle
  - Layout algorithm used for visualizing the network impacts the analysis
  - Planar structure of the triangle can not be discovered

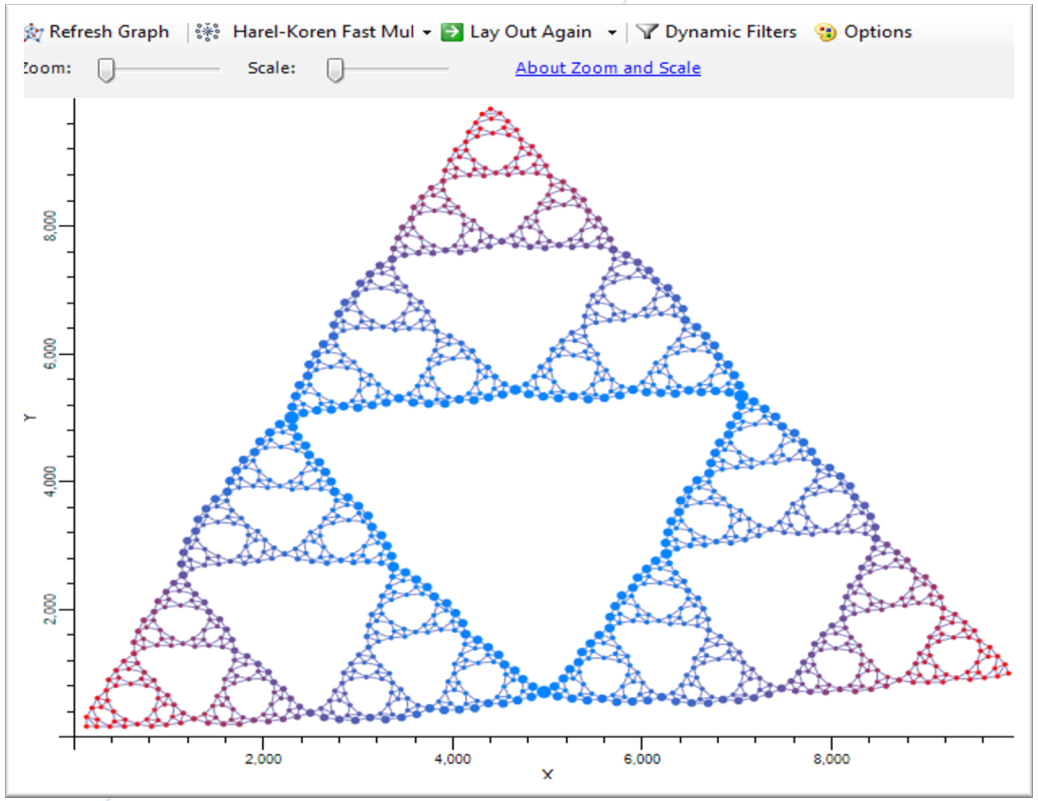


Force-directed Layout Algorithm

How good is a network visualization?

- E.g. Sierpinsky Triangle

  - Understand the tools that are applied
  - Understand the analysis methods
  - Understand the relationship between the analysis and the layout algorithm!



Harel-Koren Layout Algorithm

Information Visualization “Mantra”

- Provide and overview of the data
- Zoom and details on demand
- Dynamically filter data (e.g. nodes, links)
- Integrate metrics and visualization
- Layout through semantic substrates

Process Model for Social Network Analysis

Network Analysis and Visualization Process Model

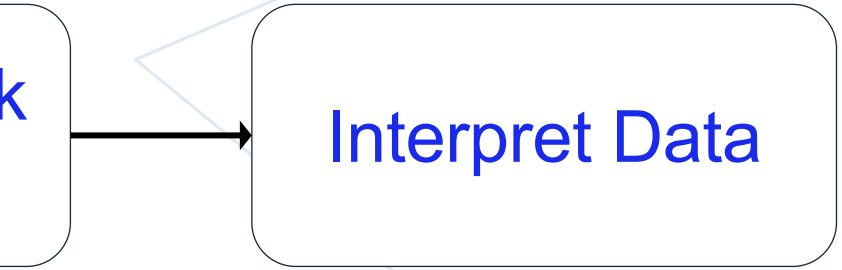
- Conceptual goal of the analysis, i.e. specifying the hypothesis and objectives
- Choices may be overwhelming, but it is important to be clear about the purpose and the insights one aims to gain
- Ideally specified before collecting the actual data, but hypotheses can also be raised against existing data sets

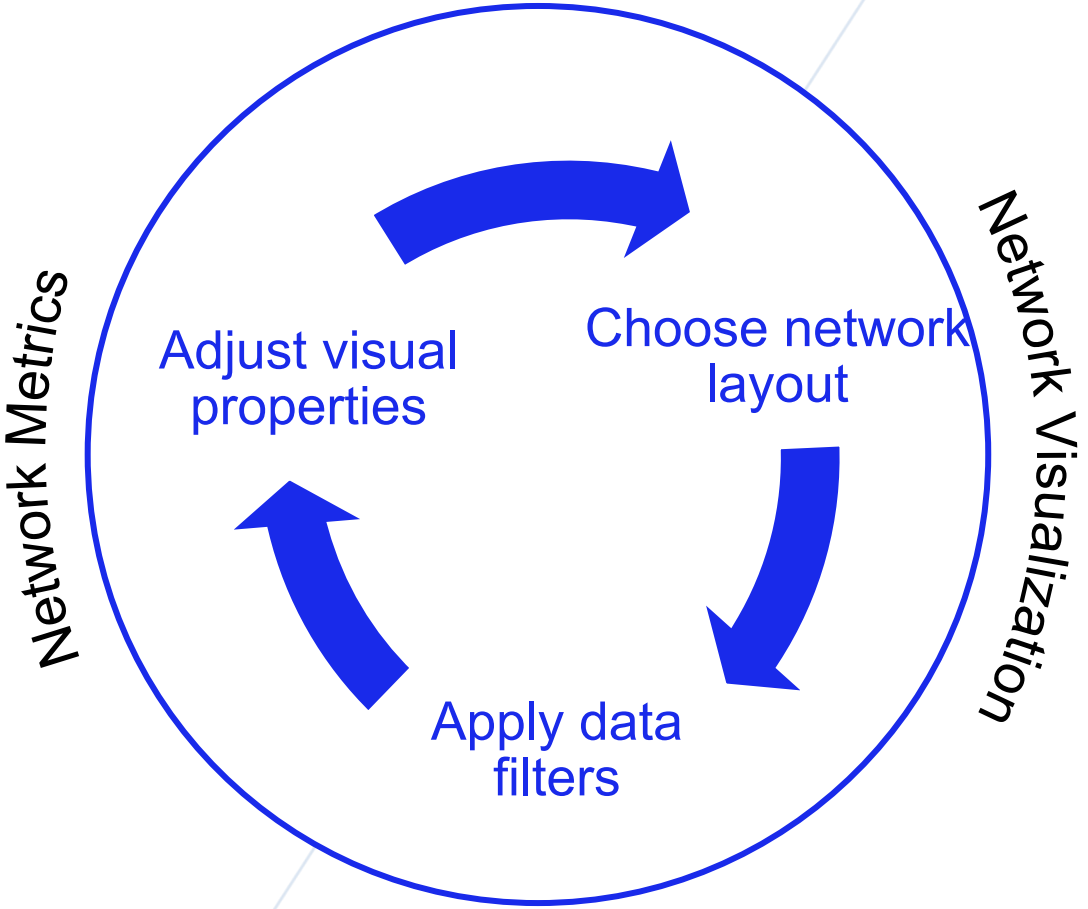
Network Analysis and Visualization Process Model

- Can be a onerous task if data is collected manually
- Automated collection via data APIs requires coding skills or specific toolkits
- May require a range of data preparation techniques, typically used in data mining
- Representation of the network data should meet the goals of the analysis

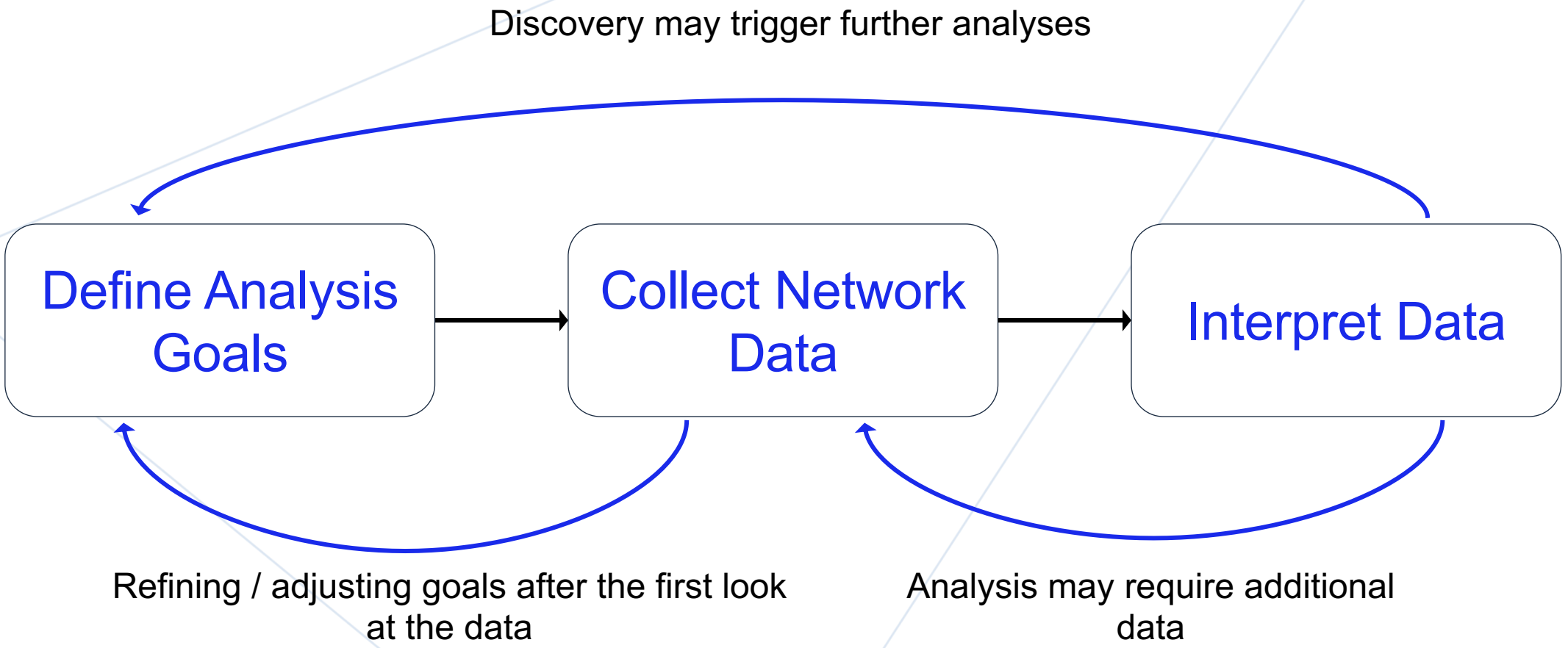
Network Analysis and Visualization Process Model

Network Analysis and Visualization Process Model





Network Analysis and Visualization Process Model



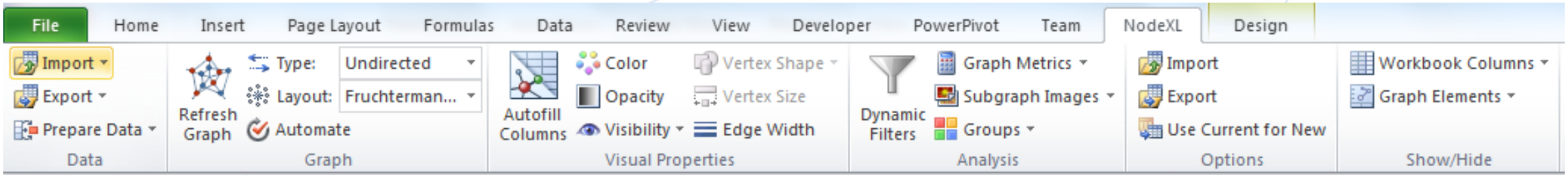

Network Analysis Toolkits

Overview

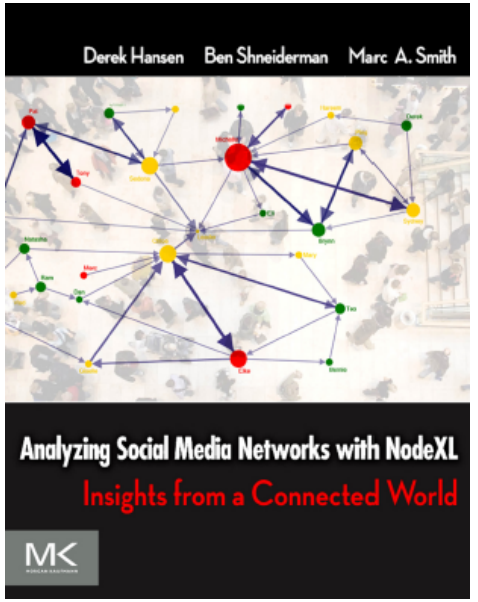
- Many software toolkits out there (see Wikipedia page)
- Which one to choose? Some considerations:
  - Supported platform (Mac, Linux, Windows)
  - Stand-alone vs. software library
  - Programming skills (none to proficient)
  - Interactive analysis and visualization
  - Built-in metrics and algorithms
  - Small/medium scale vs. large scale analysis
  - Commercial vs. open source
  - Other features: data collectors, interoperability reporting

NodeXL (https://www.smrfoundation.org/nodexl/)

Porto Business School
/ University of Porto

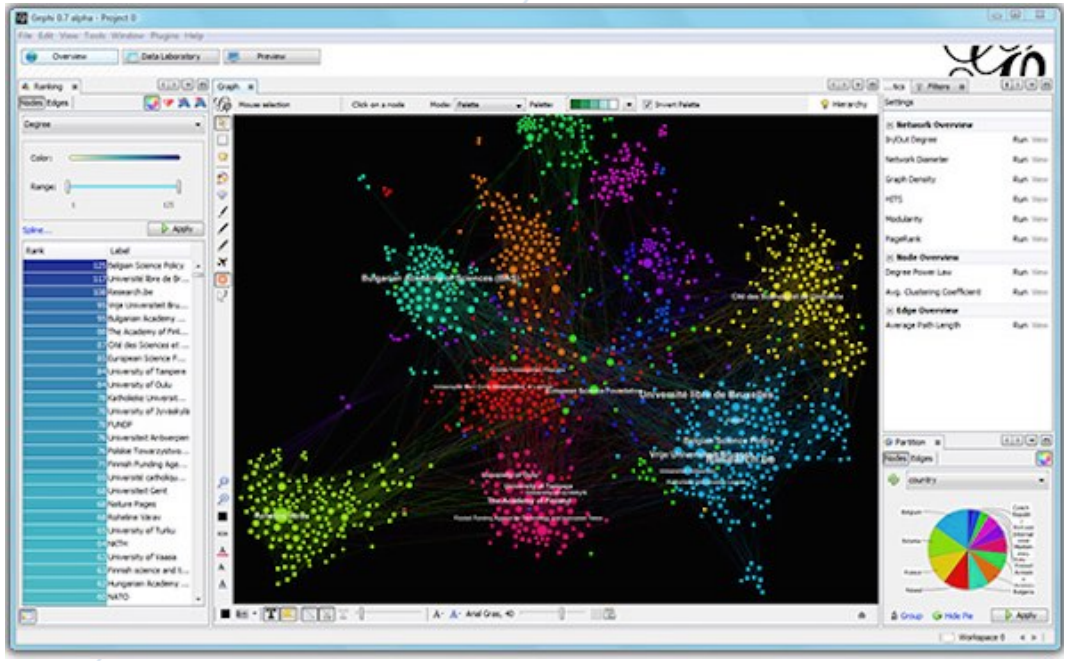


- Add-in for MS Excel makes graph theory as easy as a bar chart
- Requires MS Office / Windows version only
- Interactive visualization and exploration
- Import social networks directly from several social media services, including Twitter, Flicker, Facebook, etc.
- Flexible import and export graphs in GraphML, Pajek, UCINet, edge list and matrix formats



Gephi (http://gephi.github.io/)

- Interactive visualization and exploration
- Open source and free (Java-based)
- Runs on Windows, Linux and Mac OS X
- Supports the main file formats for networks
- Supports all kinds of networks and complex systems, dynamic and hierarchical graphs



R / igraph (http://igraph.org)

- **igraph** is a collection of network analysis tools with the emphasis on **efficiency**, **portability** and ease of use

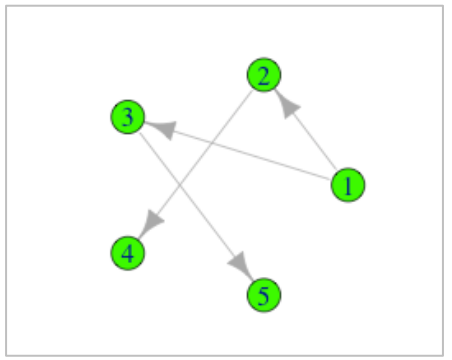
- Open source and free, can be programmed in **R**, **Python** and **C/C++**

- Runs on Windows, Linux and Mac OS

- Relies on other libraries for visualization

```
Console ~/
> library(igraph)
> g <- graph( c(1,2, 1,3, 2,4, 3,5), n=5 )
> V(g)
Vertex sequence:
[1] 1 2 3 4 5
> E(g)
Edge sequence:

[1] 1 -> 2
[2] 1 -> 3
[3] 2 -> 4
[4] 3 -> 5
> V(g)$color <- 'green'
> plot(g, layout=layout.circle, vertex.label=V(g)$name, vertex.size=30)
>
```



Network X (https://networkx.github.io/)

- Python library, relies on other libraries for visualization
- Runs on Windows, Linux and Mac OS
- Can handle large networks
- Supports the main file formats for networks
- Supports all kinds of networks and complex systems, dynamic and hierarchical graphs

```
>>> import networkx as nx
>>> g = nx.Graph()
>>> g.add_node('first node')
>>> g.add_node('second node')
>>> g.add_edge(1,2)
>>> print g.nodes()
['first node', 1, 2, 'second node']
>>> print g.edges()
[(1, 2)]
>>> g.add_edge('first node',2)
>>> print g.edges()
[('first node', 2), (1, 2)]
>>>
```


Hands-on Tutorial

Hands-on SNA Tutorial

- Define visual properties of the nodes based on attributes or *computed metrics**
- Define thickness and / or opacity of the edges based on data attributes
- Add node labels to the graph
- Interactively apply filters to analyse structural properties of the network
- Experiment with different layout algorithms
- Group nodes based on pre-defined categories or *community detection algorithm**
- Selected SNA Tools: R / igraph, NodeXL, (Gephi)

* Concepts to be introduced in Session II

Hands-on SNA with US Senate Data Set

Porto Business School / University of Porto

Define Analysis Goal

- Study 2007 US Senate voting patterns and political affinity of the senators (just after the Democrats took control)
- Make inferences about cohesiveness of the main political parties: Democrats vs. Republicans

Hands-on SNA with US Senate Data Set

Porto Business School / University of Porto

Collect Network Data

- Data set provided in 2 files
  - **US_Senate_edges.csv**
    - Edges are represented an edge list of node pairs (first 2 columns)
    - Statistics attributes about voting are included (4 columns)
      - S1, S2, Voted_same, S1_total, S2_total, Agreement, Source, Target
  - **US_Senate_nodes.csv**
    - Nodes are represented by name and have four additional attributes: the party affiliation, total votes, the US State and region in which the senator was elected
      - Senator_name, Party, Votes, State, Region, Id

Hands-on SNA with US Senate Data Set

Porto Business School / University of Porto

- Load the data in R
  - Load igraph library
  - Read CSV files

```
library(igraph)

# Define location of data files

data_path <- '<replace by path to where the data files are stored>'

# Load edges and nodes. Node index will correspond to the first column in the CSV file

e <- read.csv(paste(data_path,'US_Senate_edges.csv',sep='/'), header= TRUE)
v <- read.csv(paste(data_path,'US_Senate_nodes.csv',sep='/'), header= TRUE)

# NOTE: the file path separator in Linux/Mac is '/'; if on Windows use '\\' instead
```

Hands-on SNA with US Senate Data Set

Porto Business School / University of Porto

- Load the data in R
  - Create graph with **igraph** library
  - Add edges to graph

```
# Generate a graph from the edge list

g <- graph.edgelist(as.matrix(e[,1:2]), directed=FALSE)
print(g)
```

Hands-on SNA with US Senate Data Set

Porto Business School / University of Porto

- Add attributes to edges and nodes

```
# Assign edge attributes (column 3: 'Voted_same'; column 6: 'Agreement')

E(g)$voted_same <- e[,3]
E(g)$agreement  <- e[,6]

# Assign node attributes (column 2: 'Party'; column 4: 'Votes', column 4: 'State';
column 5: 'Region')

# Note: the index of V(g) is always be ordered. We need to make sure the order of the
attributes in the original file matches the nodes in V(g)
index <- order(v[,1])

# We now use this index to assign the node attributes. Check if the attributes are
strings or numeric and handle accordingly
V(g)$party  <- as.character(v[index,2])
V(g)$votes  <-
V(g)$state  <-
V(g)$region <-

plot(g)
```

Hands-on SNA with US Senate Data Set

Porto Business School / University of Porto

Interpret data: layouts, visual properties & dynamic filtering

- Plot the graph
- Experiment with a couple of layout algorithms and re-plot (e.g. Fruchterman-Reingold, Kamada-Kwai, Force-Atlas, Circle, Grid, etc.)
- Change the visual properties of the nodes
  - color for any of the categorical attributes
  - size for number of votes
- Dynamically filter edges based on a threshold on the percentage of votes corresponding to agreement
- Re-plot after each filtering

Hands-on SNA with US Senate Data Set

Porto Business School / University of Porto

- Interpret data in R

```
# Dynamic filter edges based on percent agreement

filter = 0.65
i = which(E(g)$agreement >= filter)

# Create subgraph of graph ‘g’, only with edges that match the above criteria
sg <- subgraph.edges(g, E(g)[i])
plot(sg)

V(g)$size <- 4
plot(sg, layout=layout.fruchterman.reingold)
```

Dynamic filter: remove edges with agreement < 0.65

Hands-on SNA with US Senate Data Set

Porto Business School / University of Porto

- Interpret data in R

```
# Color nodes based on region

colors  <- c('blue','red','black','gray','yellow','orange','pink','purple')

regions <- unique(V(sg)$region)

for (i in 1:length(regions)) {
  j = which(V(sg)$region == regions[i])
  V(sg)$color[j] <- colors[i]
}

plot(sg, layout=layout.fruchterman.reingold)
```

Nodes colour: assign different colour to senators from different regions

Hands-on SNA with US Senate Data Set

Porto Business School / University of Porto

- Interpret data in R

```
# Color nodes based on party affiliation
```

Nodes colour: assign different colour to senators from different parties

Hands-on SNA with Les Miserables graph

Porto Business School / University of Porto

- Repeat the analysis with the LesMis.gml graph

```
library(igraph)

# Define location of data files
data_path <- '<replace by path to where the gml file is stored>'

# Load GML graph
g <- read_graph(paste(data_path,'lesmis.gml',sep='/'), format = c("gml"))

# Inspect attributes of nodes and edges
vertex_attr(g)
edge_attr(g)
```

Dynamic filter: filter edges based on number of scenes both characters appeared together

References

Publications

- C. Dunne and B. Shneiderman, 2009. Improving graph drawing readability by incorporating readability metrics: A software tool for network analysts. University of Maryland, HCIL Tech Report HCIL-2009-13.
- D. L. Hansen, D. Rotman, E. M. Bonsignore, N. Milic-Frayling, E. Mendes Rodrigues, M. Smith, and B. Shneiderman, “Do you know the way to SNA?: A process model for analyzing and visualizing social media data.” in University of Maryland Tech Report: HCIL-2009-17.

References

Some SNA Software Toolkits

- iGraph – a collection of network analysis tools, than can be programmed in R, Python and C/C++
- Gephi – a platform for graph analysis and visualization
- NetworkX – a Python package for analysis of the structure, dynamics, and functions of complex networks
- NodeXL – an interactive graph analysis and visualization toolkit for MS Excel 2007 / 2010 (Windows only)
- SNAP – a C++ library for working with large scale networks
- Pajek – a program for large network analysis (Windows only)

References

Some Network Data Repositories

- http://www-personal.umich.edu/~mejn/netdata/
- https://snap.stanford.edu/data/ (large scale networks)
- http://vlado.fmf.uni-lj.si/pub/networks/data/ucinet/ucidata.htm
- http://nodexlgraphgallery.org/

References

Tutorials

- iGraph tutorials:
  - https://rstudio-pubs-static.s3.amazonaws.com/74248_3bd99f966ed94a91b36d39d8f21e3dc3.html
  - http://igraph.wikidot.com/r-tutorial
- NodeXL Teaching Resources:
  - http://www.smrfoundation.org/nodexl/teaching-with-nodexl/teaching-resources/
- Learn how to use Gephi: http://gephi.github.io/users/

Make change happen