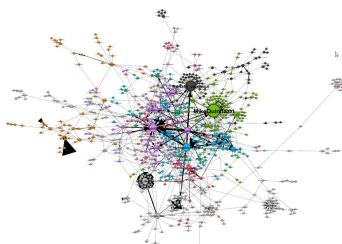




# Department of Information Systems

## School of Business & Economics



### PhD Course on Social Network Analysis

Prof. Dr. Daniel Fürstenau  
20 February 2017

[https://github.com/dfurstenau/SNA\\_class](https://github.com/dfurstenau/SNA_class)

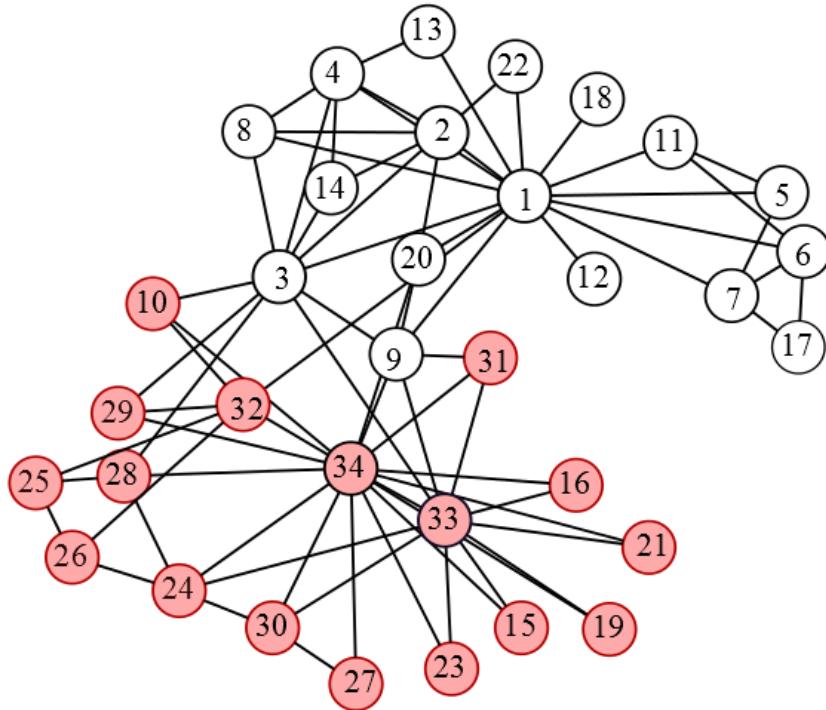
# Agenda

---

1. Introduction and examples
2. Fundamental concepts
  1. Creating graphs
  2. Visualizing graphs
  3. Centrality
  4. Describing the macro structure
  5. Community detection
3. Getting and handling data
4. Extended concepts
5. Discussion and conclusion

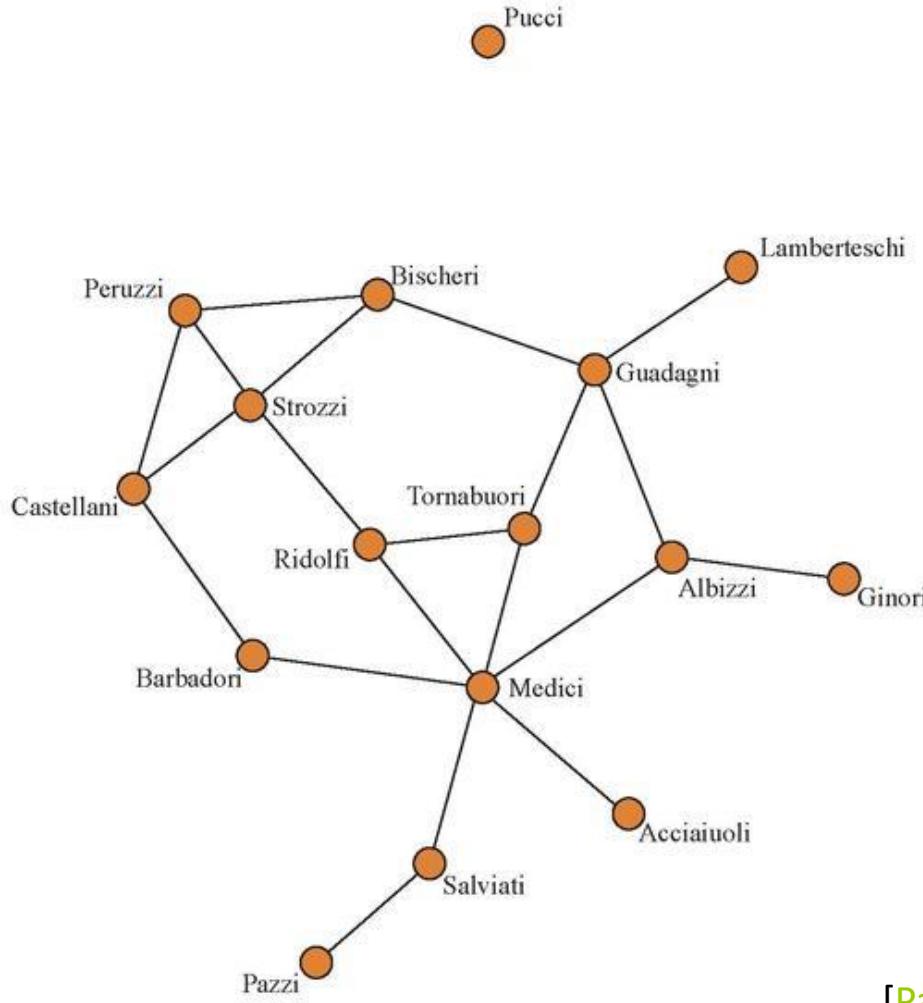
# Introduction 1: Zachary's Karate Club Network

*The Iris dataset of social network analysis*



A social network of a karate club was studied by Wayne W. Zachary for a period of three years from 1970 to 1972. The network captures 34 members of a karate club, documenting 78 pairwise links between members who interacted **outside** the club. During the study a conflict arose which led to the split of the club into two. Based on collected data Zachary assigned correctly all but one member of the club to the groups they actually joined after the split.

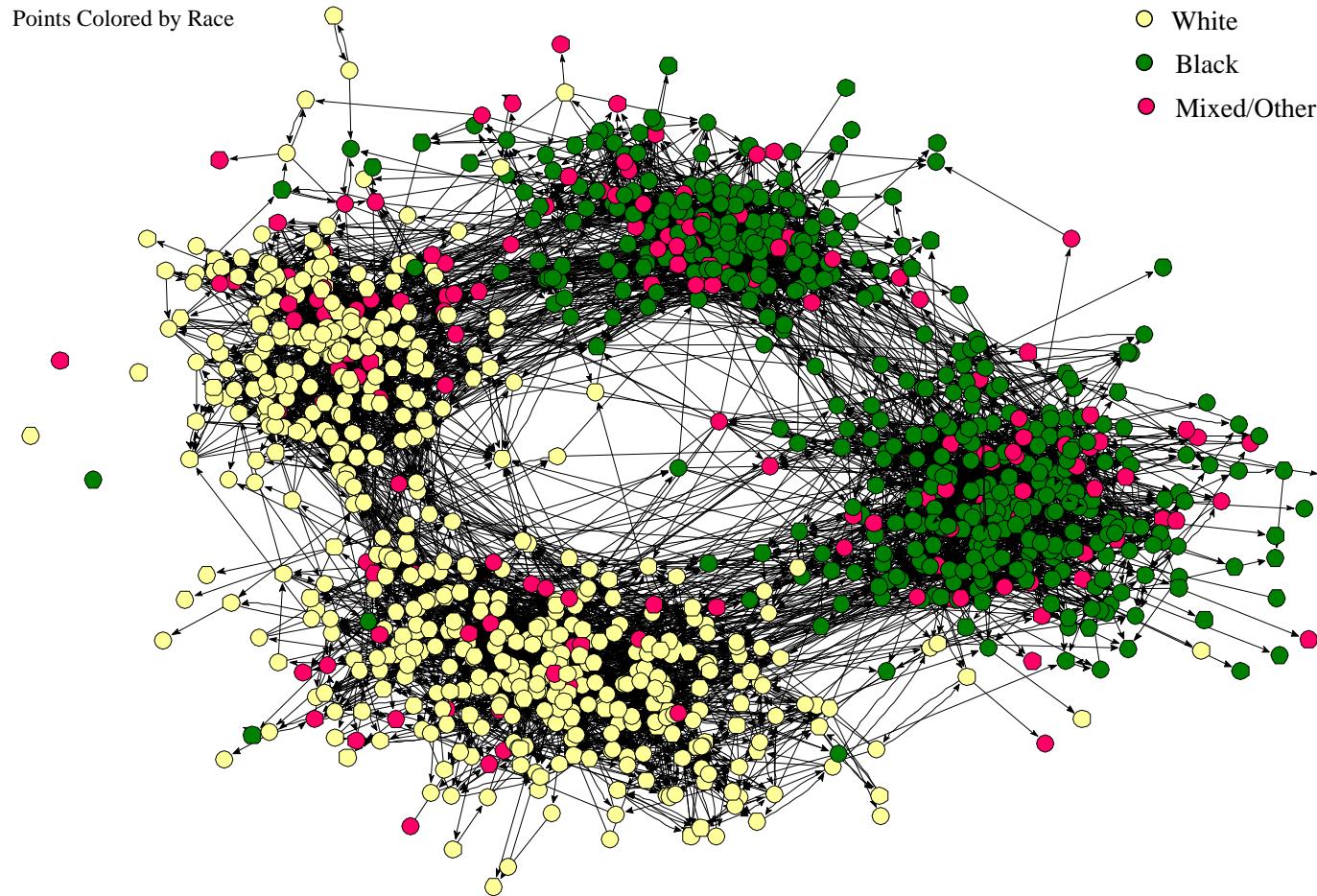
# Introduction 2: 15th Century Florentine Marriages



[Padgett and Ansell, 1993]

# Introduction 3: Add Health School Friendship Network

## The Social Structure of “Countryside” School District

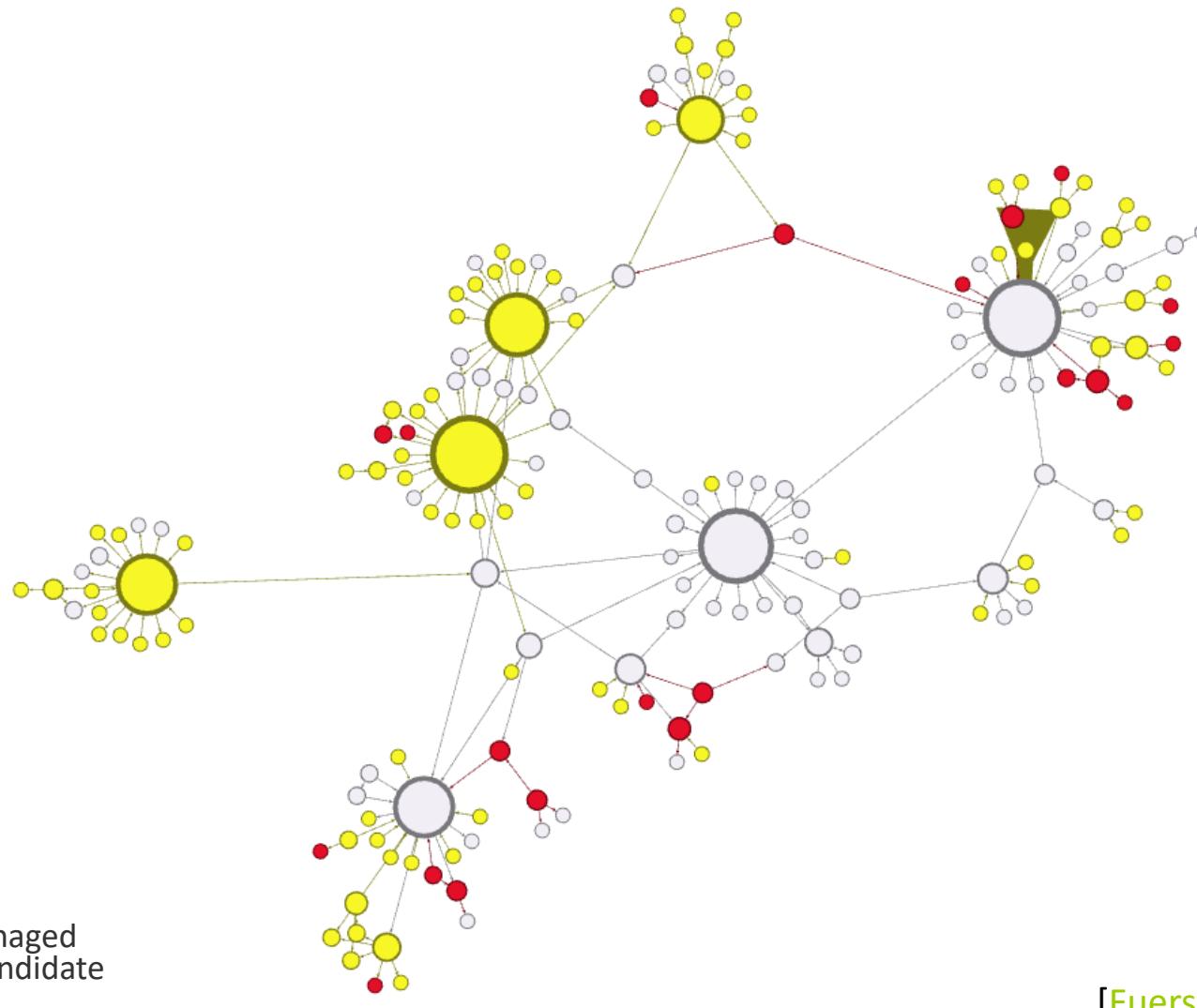


# Introduction 4: Retweet network from Twitter



[<https://www.slideshare.net/giladlotan/pydata-gilad-networkxupload>]

# Introduction 5: Recyle Inc. Information Infrastructure Network



○ Centrally managed  
● Shadow IT candidate  
■ Shadow IT

[Fuerstenau et al. 2014]

# Definitions



**Network:** a pattern of interconnections among a set of things

**Social Network:** a network where the things are people and the interconnections are social interactions

**Social Network Analysis (SNA):** the application of graph and network theory / network science to investigate social structures.

**Graph theory:** a branch of mathematics concerned with modeling pairwise interactions among objects

**Network theory:** Theorizing networks includes at least: individual elements; pairwise relationships between those elements; and a global or macro- patterning that can be considered as network structure (Brandes et al. 2013)

**Network science:** The interdisciplinary, empirical, data-driven, quantitative, computational study of network models (view of Laslo Barabasi).

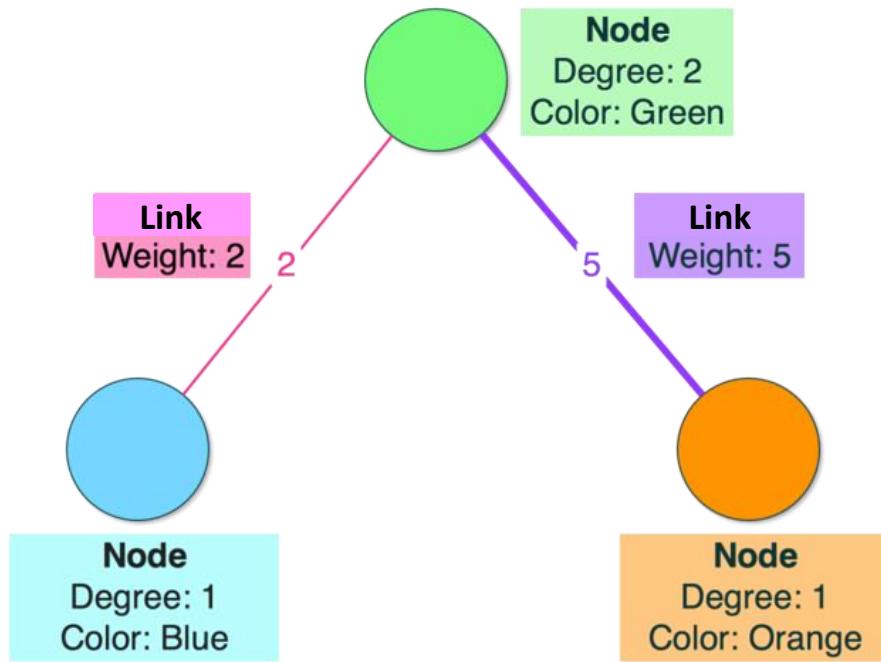
# Types of social interactions



Similarities			Social Relations					Interactions	Flows
Location	Membership	Attribute	Kinship	Other role	Affective	Cognitive	e.g.,	e.g.,	
e.g., Same spatial and temporal space	e.g., Same clubs Same events etc.	e.g., Same gender Same attitude etc.	e.g., Mother of Sibling of	e.g., Friend of Boss of Student of Competitor of	e.g., Likes Hates etc.	e.g., Knows Knows about Sees as happy etc.	Sex with Talked to Advice to Helped Harmed etc.	Information Beliefs Personnel Resources etc.	

[Borgatti et al. 2009]

# Parts of a network: Nodes and links



**Node / Vertex:** The entity of analysis which has a relationship. Node is used in the network context, vertex is used in the graph theory context, but both terms are often used interchangeably.

**Link / Edge / Relationship:** The connections between the nodes. Link is used in the network context, edge is used in the graph theory context, and all words are used interchangably with *relationship*.

**Attributes:** Both nodes and links can store attributes, which contain additional data about that object.

**Weight:** A common *attribute* of links, used to indicate *strength* or *value* of a relationship.

**Degree:** Number of links a node has.

# Types of networks

Networks are typically classified based on the presence of **weights** and **direction** attached to the links in a graph. The table below covers what we call each type of graph:

	Absent	Present
Weights	Unweighted	Weighted
Directionality	Undirected	Directed

**Additional flavors:** parallel links, self-loops,  $n$ -partite graphs

In this context:

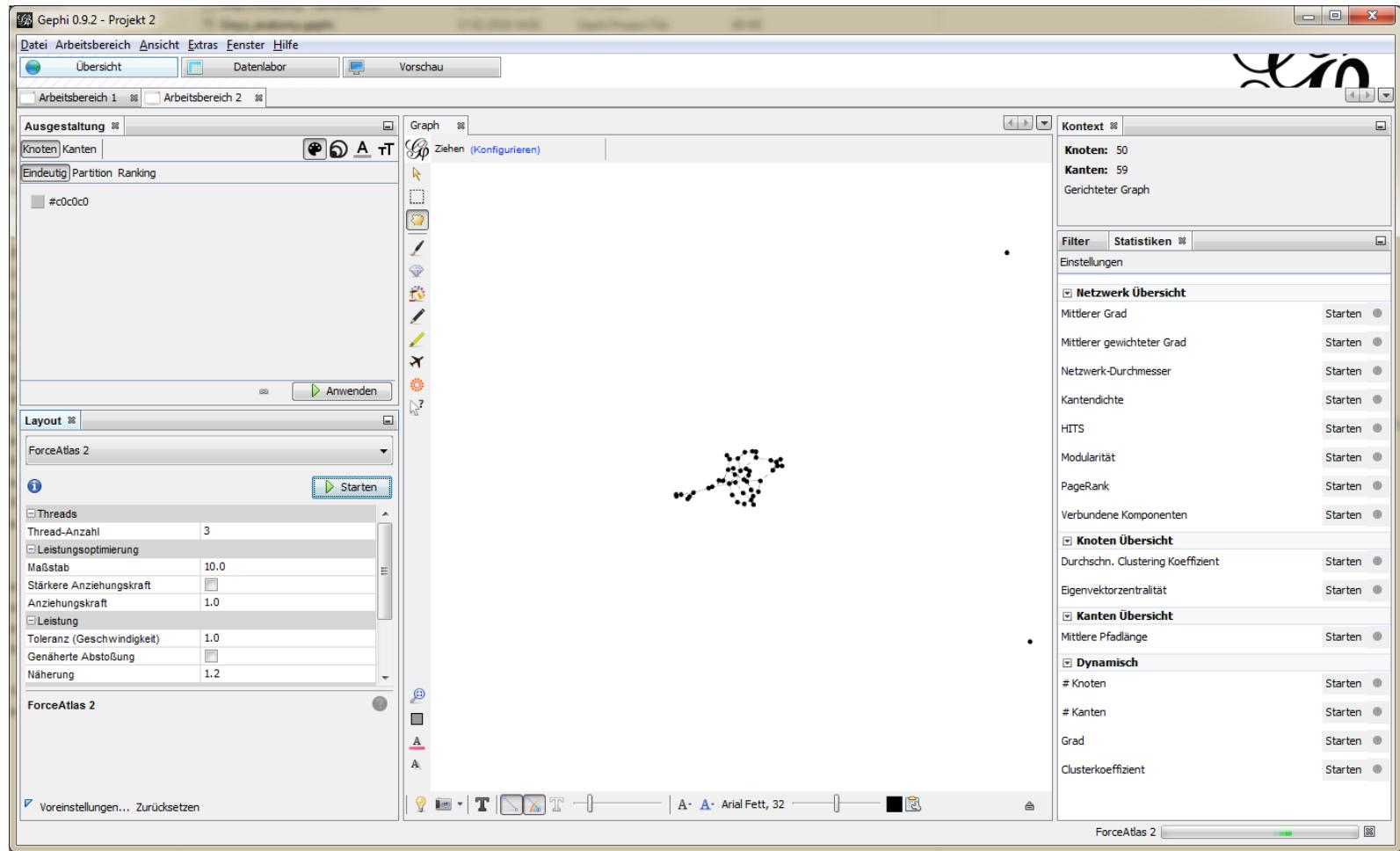
We are talking about a(n) [unweighted/weighted] [undirected/directed] graph (with [parallel edges | self loops]).

# Agenda

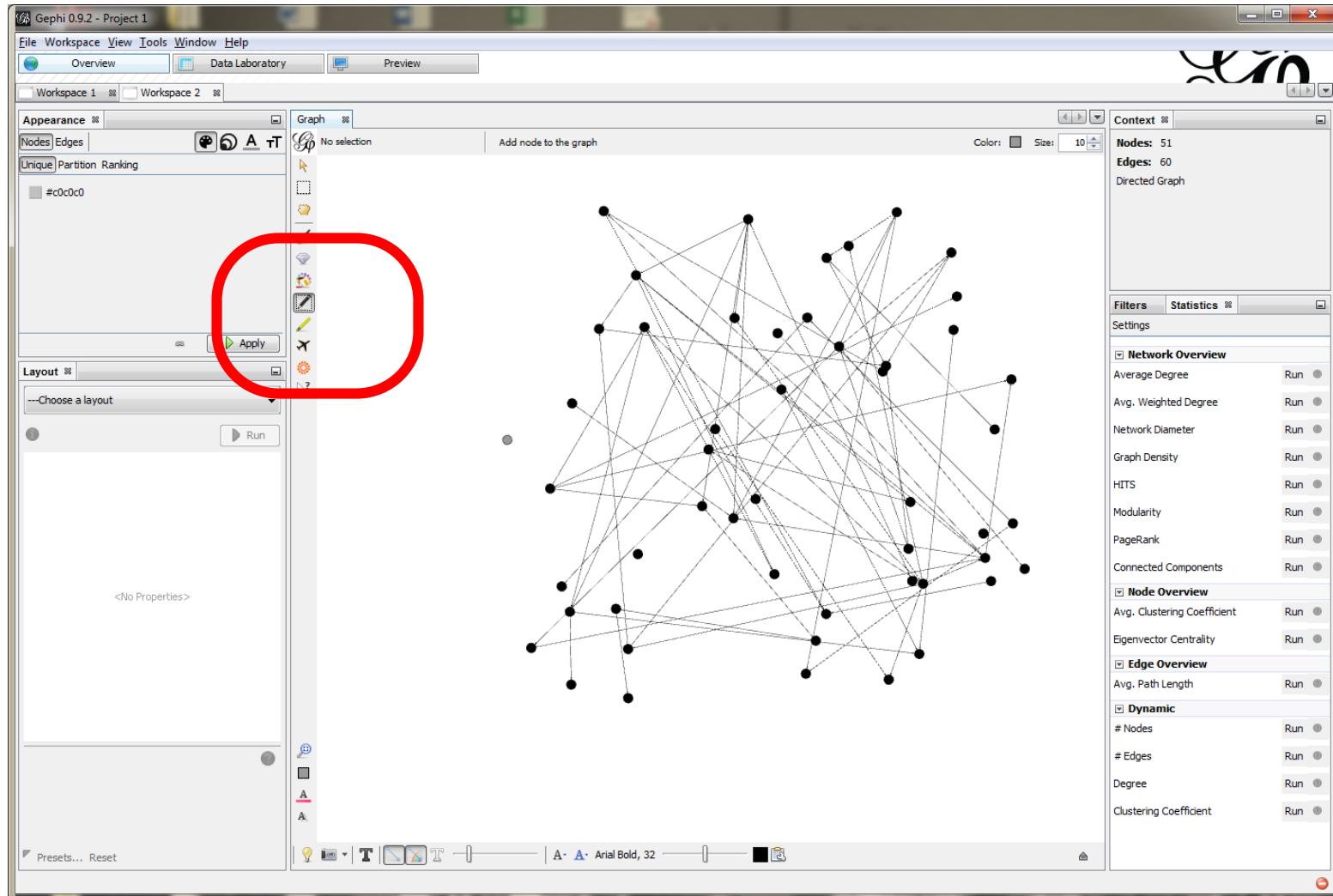
---

1. Introduction and examples
2. Fundamental concepts
  1. Creating graphs
  2. Visualizing graphs
  3. Centrality
  4. Describing the macro structure
  5. Community detection
3. Getting and handling data
4. Extended concepts
5. Discussion and conclusion

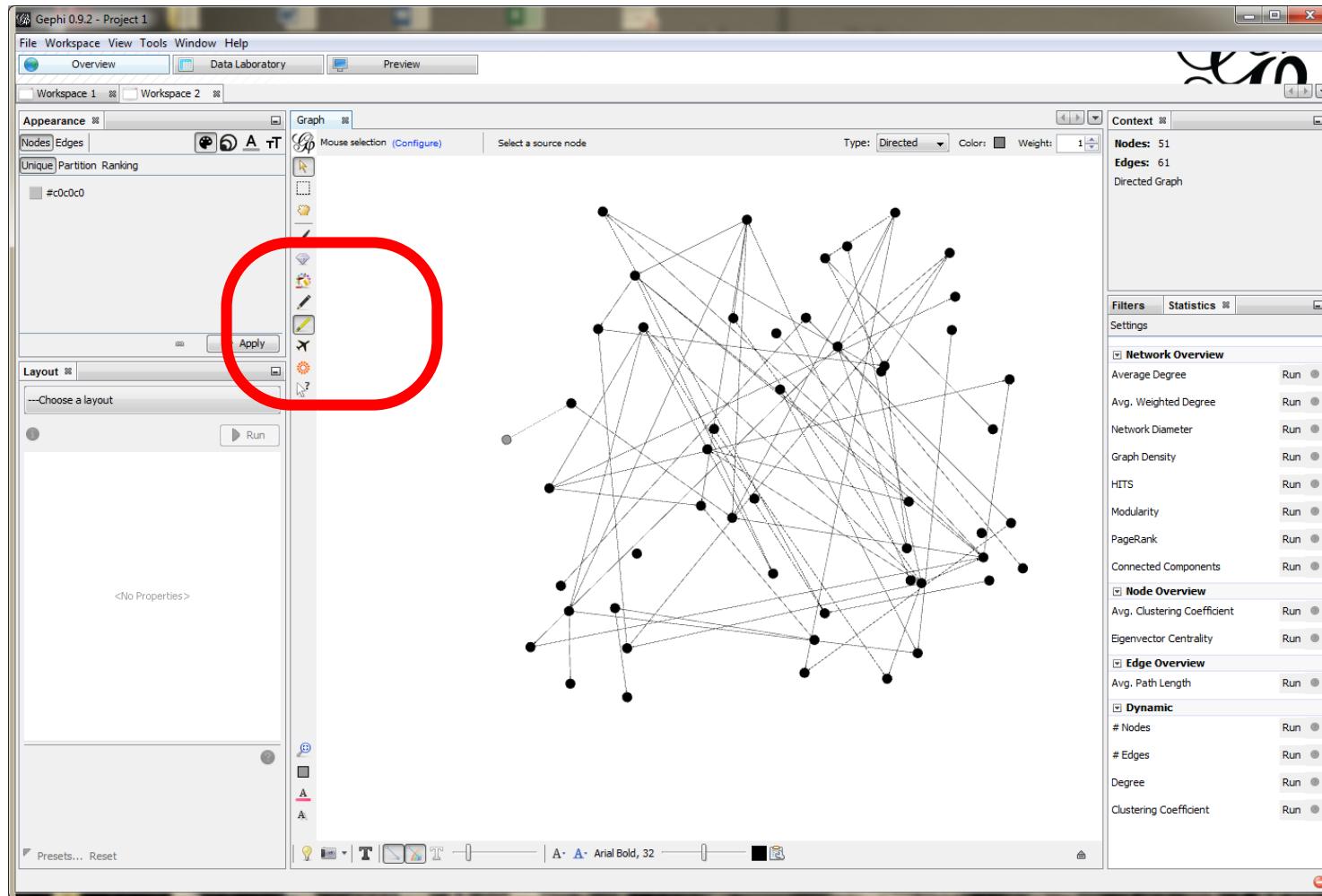
# Creating a graph... in Gephi



# Adding nodes...



# Adding links...



# Adding and inspecting attributes...

The screenshot shows the Gephi 0.9.2 interface with the 'Data Laboratory' tab selected. A 'Data Table' window displays a list of edges with columns for Source, Target, Type, Id, Label, Interval, and Weight. An 'Add column - Settings' dialog is open over the table, prompting for a 'Title' (which is empty) and a 'Type' (set to 'String'). A validation error message 'Column title can't be empty' is displayed. Below the table, a toolbar contains icons for various data manipulation operations like 'Add column', 'Merge columns', and 'Convert column to dynamic'.

Source	Target	Type	Id	Label	Interval	Weight
1	10	Directed	0	0		1.0
1	11	Directed	1	1		1.0
1	27	Directed	2	2		1.0
1	28	Directed	3	3		1.0
1	31	Directed	4	4		1.0
1	32	Directed	5	5		1.0
1	42	Directed	6	6		1.0
3	8	Directed	7	7		1.0
3	40	Directed	8	8		1.0
4	15	Directed	9	9		1.0
4	32	Directed				1.0
4	46	Directed				1.0
4	47	Directed				1.0
5	17	Directed				1.0
6	42	Directed				1.0
7	22	Directed				1.0
7	29	Directed				1.0
8	12	Directed				1.0
8	19	Directed				1.0
8	24	Directed				1.0
8	34	Directed				1.0
9	37	Directed				1.0
9	46	Directed	22	22		1.0
10	21	Directed	23	23		1.0
11	12	Directed	24	24		1.0
11	36	Directed	25	25		1.0
12	40	Directed	26	26		1.0
13	27	Directed	27	27		1.0
14	38	Directed	28	28		1.0
16	41	Directed	29	29		1.0
17	25	Directed	30	30		1.0
17	31	Directed	31	31		1.0
17	35	Directed	32	32		1.0
17	41	Directed	33	33		1.0
17	46	Directed	34	34		1.0

This is to be done in the data laboratory.



# Reading in different representations of graphs

Data for graphs and networks comes in many different representations.

- ▶ **Representations:**

- ▶ Link list
- ▶ Adjacency matrix
- ▶ ...

**Note:** representations are related to, but distinct from, the storage format.  
We will use `csv`-files for link lists and `.dl` for adjacency matrices.

There are several other formats that we briefly touch upon such as `.gexf`,  
`.graphml`, and `.gephi`.

# Grey's anatomy example data set



The dataset we'll look at is a record of all "romantic" encounters between characters on the TV show Grey's Anatomy

[[Rob Chew and Peter Baumgartner](#)]

# Edge lists

An edge list (aka link list) is a common way of representing a graph. This representation can be thought of as a list of tuples, where each tuple represents an edge between two of the nodes in your graph. The nodes of the graph can be inferred by taking the set of objects from all tuples.

You can infer/determine whether a graph is directed or weighted from an edge list.

- **Weighted:** If edges appear more than once, or if an additional weight attribute is added as a 3rd column, the graph is weighted
- **Directed:** If the "From" and "To" (often seen as "Source" and "Target") of an edge in the list is not arbitrary, it's a directed graph

	A	B	C
1	Source	Target	Type
2	lexi	sloan	Undirected
3	lexi	karev	Undirected
4	owen	yang	Undirected
5	owen	altman	Undirected
6	sloan	torres	Undirected
7	sloan	altman	Undirected
8	torres	arizona	Undirected
9	torres	karev	Undirected
10	derek	grey	Undirected

01-Greys\_anatomy\_edgelist.csv

# Adjacency matrices

A common way of representing graph data is through an adjacency matrix -- often referred to mathematically as  $A$ . This data structure is a square,  $n \times n$  matrix where  $n = \text{number of nodes}$ . Each column and row in the matrix is a node. For any two nodes,  $i$  and  $j$  the value at  $A_{ij}$  (row  $i$  and column  $j$ ) represents the weight of the link between nodes  $i$  and  $j$ .

	denny	kepner	grey	colin	finn
denny	0.0	0.0	0.0	0.0	0.0
kepner	0.0	0.0	0.0	0.0	0.0
grey	0.0	0.0	0.0	0.0	1.0
colin	0.0	0.0	0.0	0.0	0.0
finn	0.0	0.0	1.0	0.0	0.0

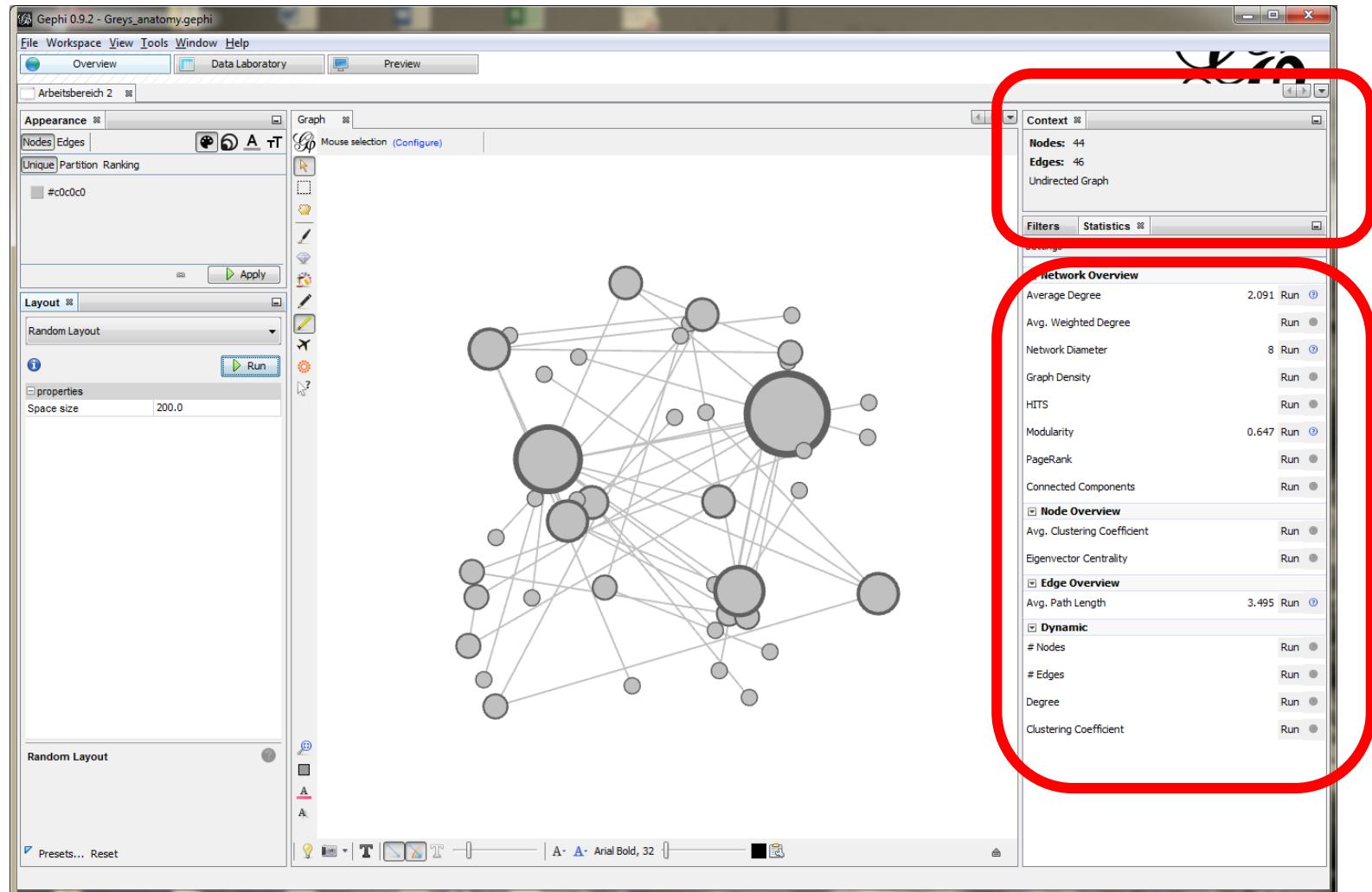
# Adjacency matrices (continued): DL format

Greys\_anatomy\_adjacency\_matrix.dll X

```
dl n=44
format = fullmatrix
labels:
addison,adele,altman,amelia
data:
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0
0 0 0 0 0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0
```

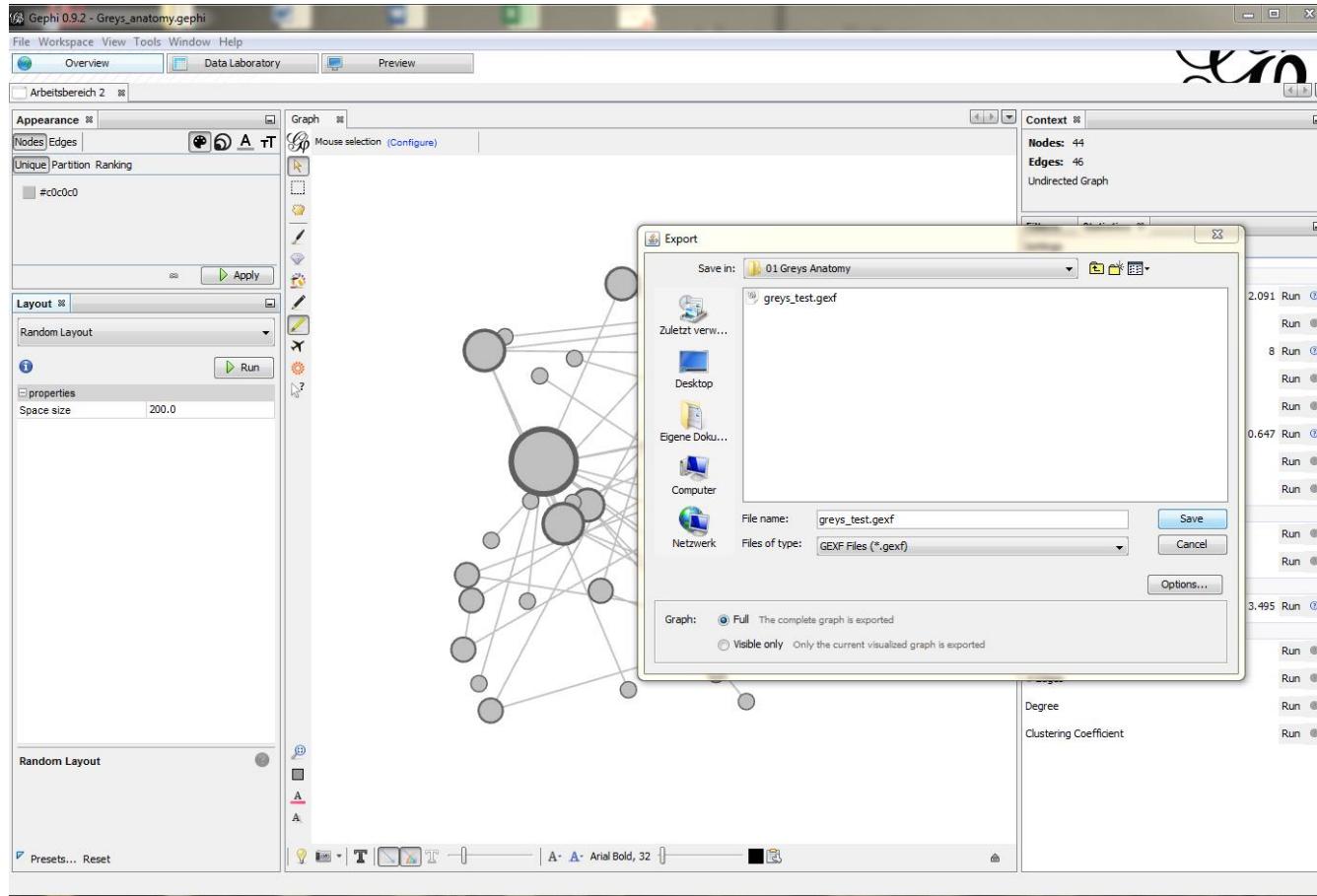
Greys\_anatomy\_adjacency\_matrix.dll

# Summary statistics on graph...



# Exporting graphs

We'll export the graph in GEXF (Graph Exchange XML Format).



# Exporting: Where to?

---

- ▶ **Analysis in other developer tools and packages**
  - ▶ Iphyton/Jupyter and NetworkX (my package of choice)
  - ▶ Further python packages: igraph, D3 (visualization)
  - ▶ R (SNA package etc.)
  - ▶ ...
- ▶ **Analysis in other packaged software programs**
  - ▶ Ucinet
  - ▶ Pajek
  - ▶ Cytoscape
- ▶ ...

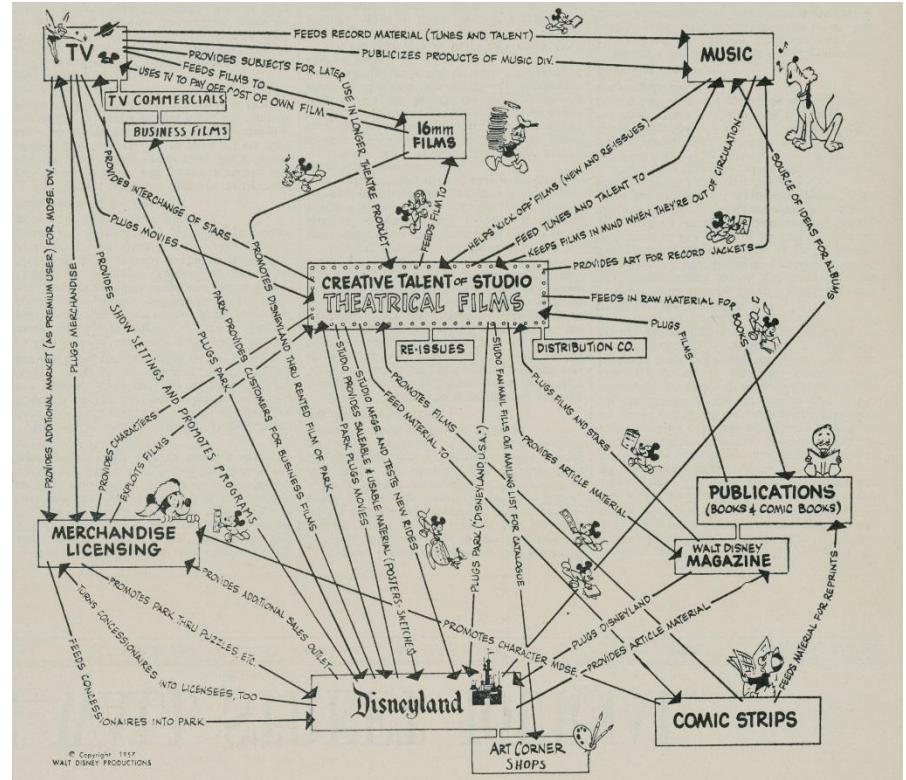
# Agenda

---

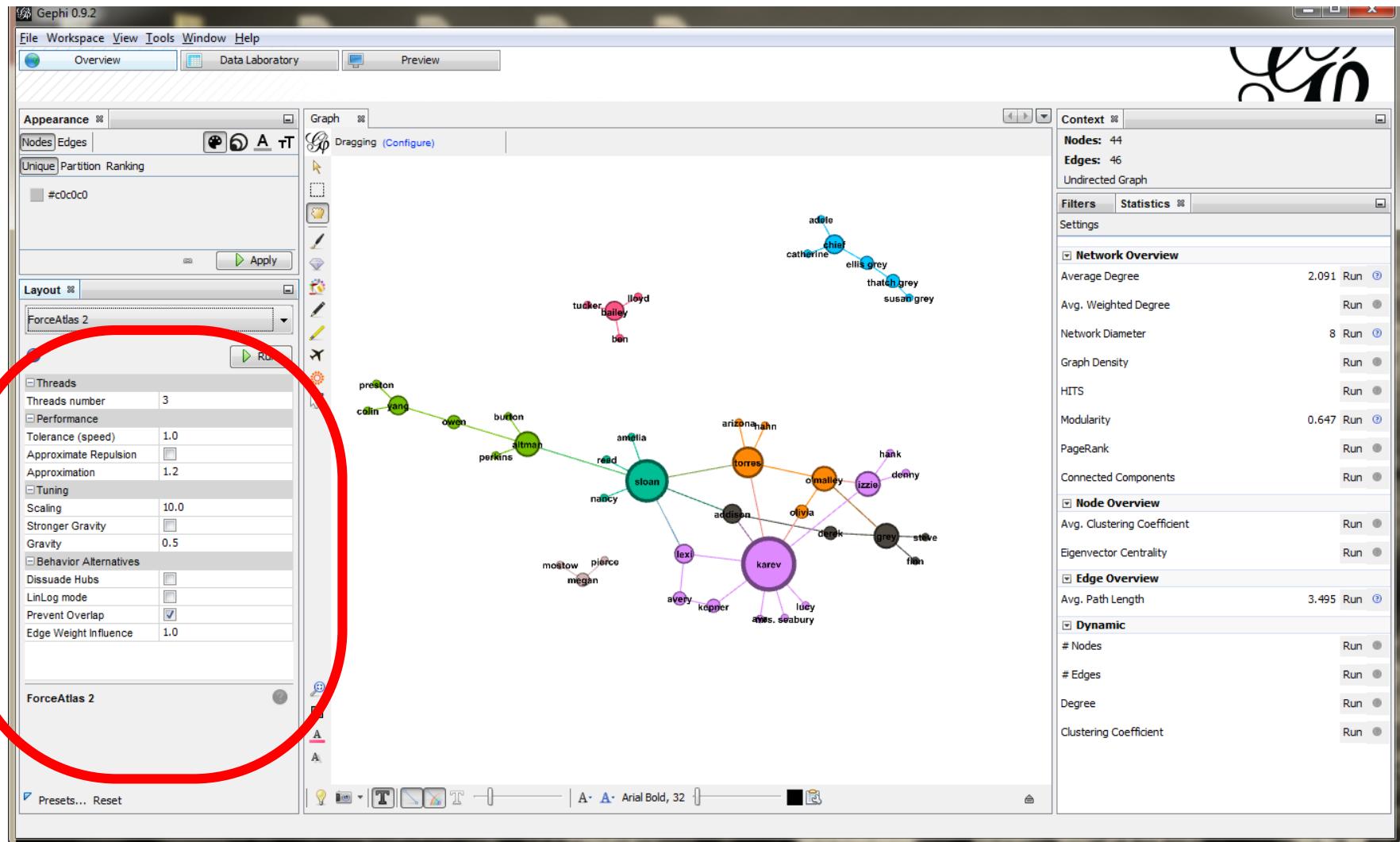
1. Introduction and examples
2. Fundamental concepts
  1. Creating graphs
  2. **Visualizing graphs**
  3. Centrality
  4. Describing the macro structure
  5. Community detection
3. Getting and handling data
4. Extended concepts
5. Discussion and conclusion

# Layout algorithms

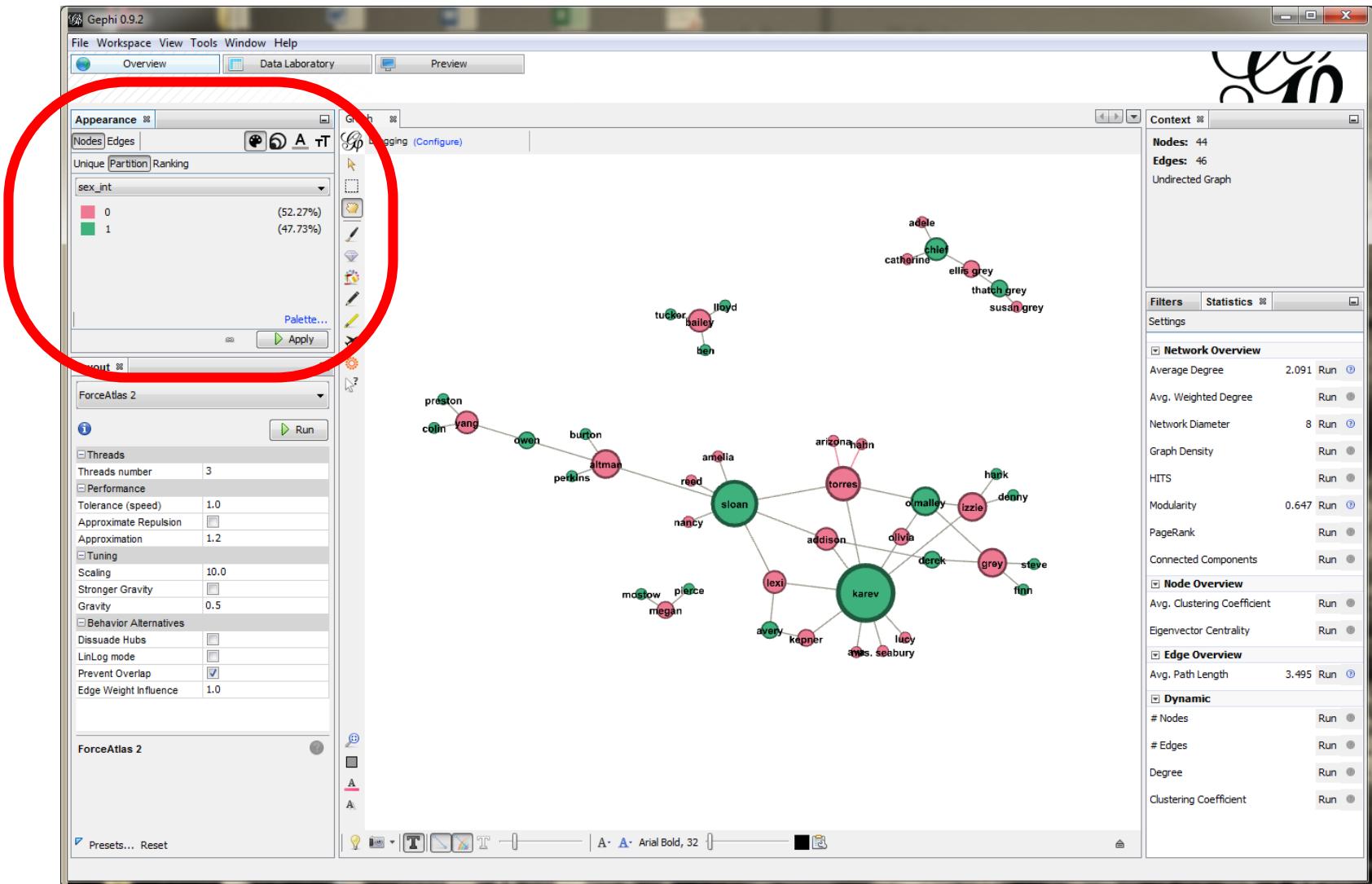
- Vizualizing networks is a complicated problem -- how do you position the nodes and edges in a way such that no nodes overlap, connected nodes are near each other, none of the labels overlap? Typically we use what is called a *layout* to plot or visualize networks. A layout is an algorithm used to position nodes and edges on a plot automatically in aesthetically and informationally satisfactory ways.
  - There are several different layout algorithms, but the most common is a *force-directed* layout. These layout algorithms are based off of physical repulsion and spring systems. In general, the rule for force-directed layouts is: repel all nodes, and model connections between nodes as 'springs', with the result that more connected nodes will be closer together.
  - One important issue is that each layout typically has random initial conditions. Running a plot function twice will return two different plots, both following the rules of the algorithm, but differing due to the initial conditions of the layout.



# Layout algorithms



# Detailed plotting with colors by attribute



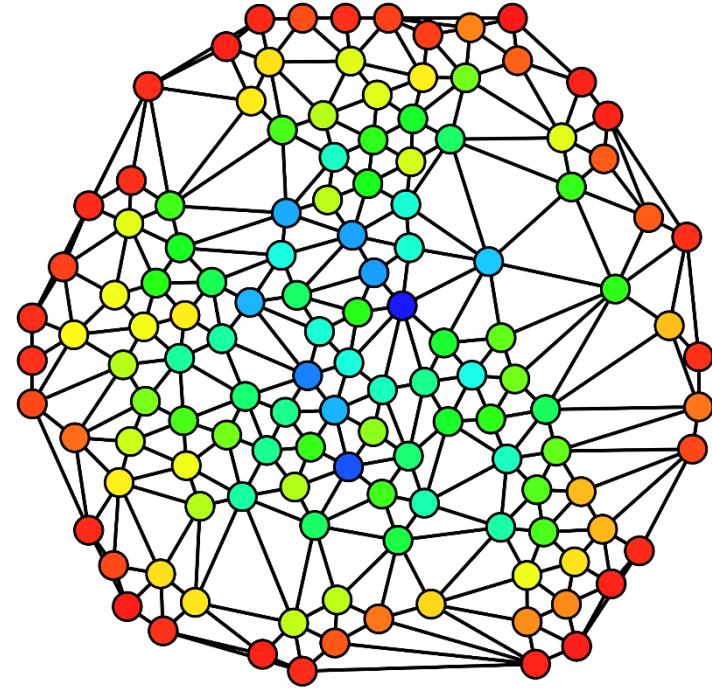
# Agenda

---

1. Introduction and examples
2. Fundamental concepts
  1. Creating graphs
  2. Visualizing graphs
  3. **Centrality**
  4. Describing the macro structure
  5. Community detection
3. Getting and handling data
4. Extended concepts
5. Discussion and conclusion

# Centrality measures – overview

- Definition of centrality
- Compare and contrast popular centrality measures on dataset
  - Degree
  - Closeness
  - Betweenness
  - Eigenvector



# Degree centrality

The **degree** of a node is the number of other nodes to which it is connected.

Gephi's degree centrality is calculated as the absolute value. Other programs are taking the degree of the node and dividing by  $n-1$  where  $n$  is the number of nodes in  $G$ .

$$C_D(u) = \frac{deg(u)}{n - 1}$$

**NOTE:** In a directed graph, both in-degree and out-degree centrality can be calculated.

Let's find the degree of our main character Grey (4).

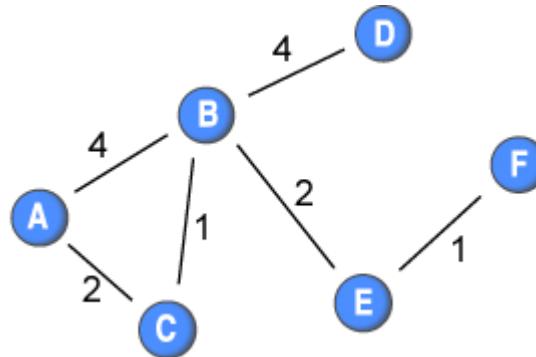
# Likewise we find the degree of each cast member...

The screenshot shows the Gephi 0.9.2 interface with the title "Gephi 0.9.2 - Greys\_anatomy.gephi". The main window displays a "Data Table" with three columns: "Id", "Label", and "Grad". The "Label" column lists character names, and the "Grad" column shows their degree values. The row for "grey" has a blue highlight, indicating it is selected. The bottom of the table has various data manipulation buttons.

Id	Label	Grad
22	karev	9
38	sloan	7
42	torres	5
3	altman	4
18	grey	4
21	izzie	4
32	o'malley	4
1	addison	3
8	bailey	3
12	chief	3
24	lexi	3
44	yang	3
7	avery	2
15	derek	2
16	ellis grey	2
23	kepner	2
27	megan	2
31	olivia	2
33	owen	2
41	thatch grey	2
2	adele	1
4	amelia	1
5	arizona	1
6	ava	1
9	ben	1
10	burton	1
11	catherine	1
13	colin	1
14	denny	1
17	finn	1
19	hahn	1
20	hank	1
25	lloyd	1
26	lucy	1
28	mostow	1

# Closeness centrality

Closeness Centrality measures how many "hops" it would take to reach every other node in a network (taking the shortest path). It can be informally thought as 'average distance' to all other nodes.



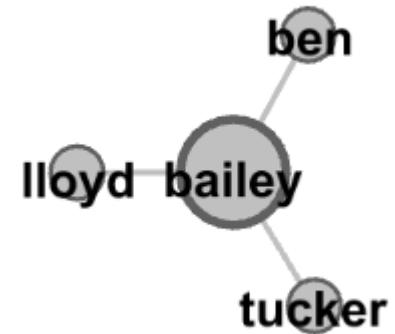
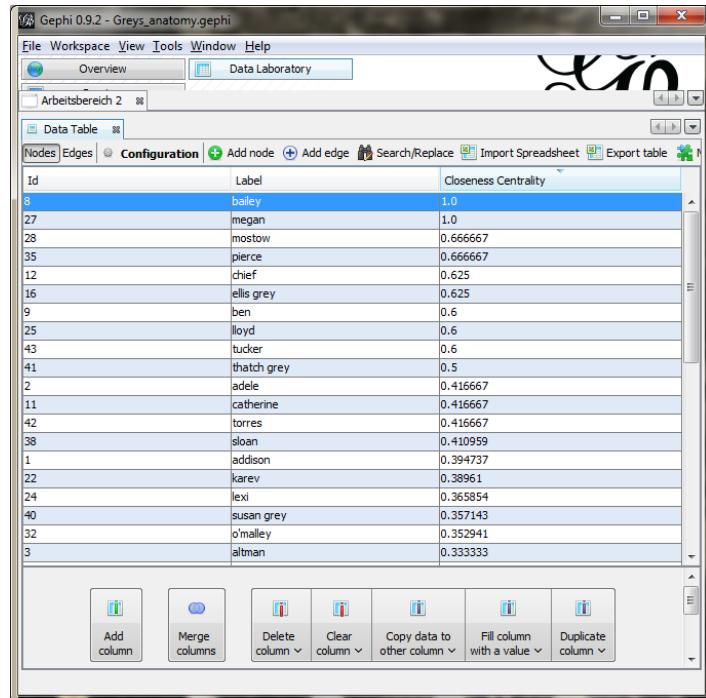
In Gephi, it is the reciprocal of the *average* value, which normalizes the value in a 0 to 1 range.

$$C_C(u) = \frac{n - 1}{\sum_{v=1}^{n-1} d(v, u)}$$

# Closeness centrality

$$C_C(u) = \frac{n - 1}{\sum_{v=1}^{n-1} d(v, u)}$$

Why has Bailey a value of 1?



$$\begin{aligned} & 4 - 1 \\ & \hline \\ & 1+1+1 \end{aligned}$$

**NOTE:** If the graph is not completely connected, this algorithm computes the closeness centrality for each connected part separately.

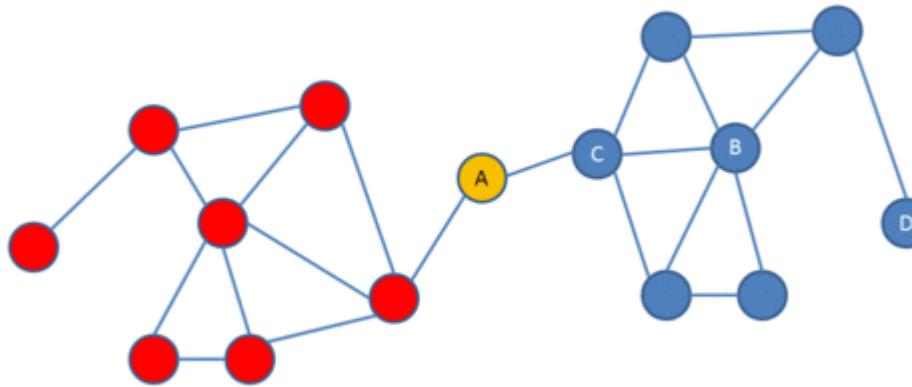
# Why care about closeness centrality?

---

- Degree centrality measures might be criticized because they only take into account the immediate ties that an actor has, or the ties of the actor's neighbors, rather than indirect ties to all others.
- One actor might be tied to a large number of others, but those others might be rather disconnected from the network as a whole.
- In a case like this, the actor could be quite central, but only in a local neighborhood.

# Betweenness centrality

Betweenness centrality quantifies the number of times a node acts as a bridge (or "broker") along the shortest path between two other nodes.



In this conception, nodes that have a high frequency to occur on a shortest path between two nodes have a high betweenness.

$$C_B(v) = \sum_{s,t \in V} \frac{\sigma(s, t|v)}{\sigma(s, t)}$$

where  $\sigma(s, t)$  is total number of shortest paths from node  $s$  to node  $t$  and  $\sigma(s, t|v)$  is the number of those paths that pass through  $v$ .

# Why care about betweenness centrality?

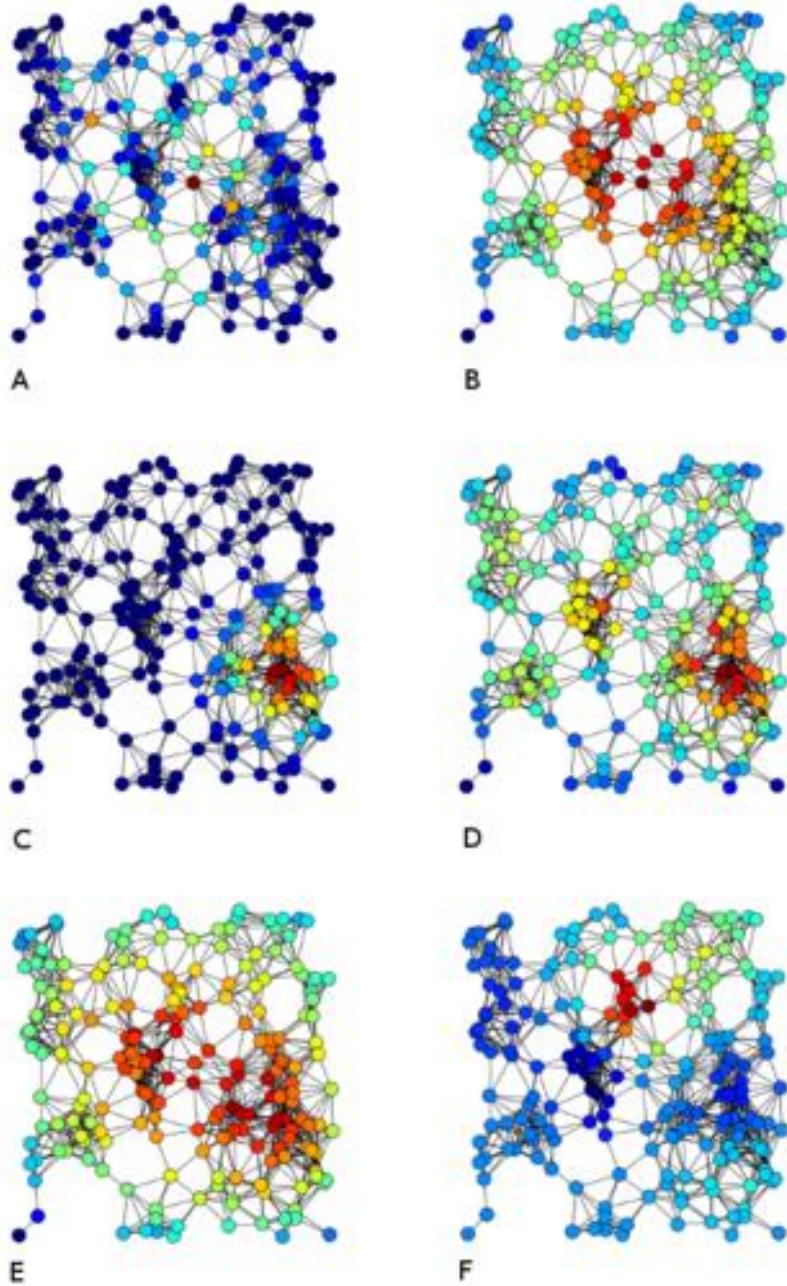
---

- ▶ Intermediaries occupy powerful positions
  - ▶ Boundary spanning and nexus work – connecting different subgraphs
- 
- ▶ Interpretation: Suppose you need permission from ex-partners to date. How many persons would Preston need to date Torres?

# Eigenvector centrality

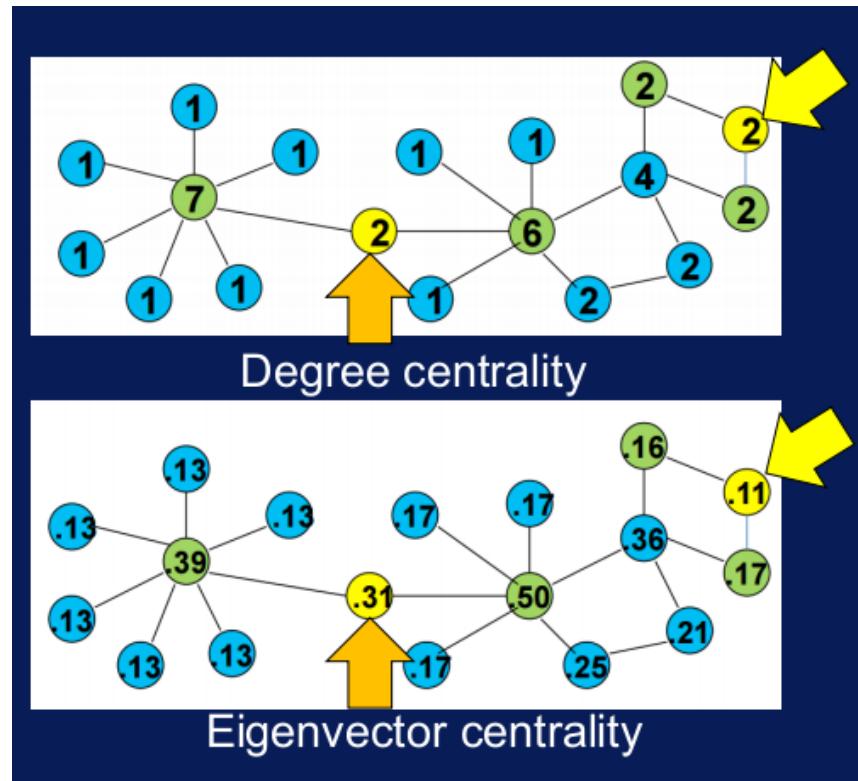
A node is high in eigenvector centrality if it is connected to many other nodes who are themselves well connected. Eigenvector centrality for each node is calculated as the proportional eigenvector values of the eigenvector with the largest eigenvalue.

**Middle Left ("C"):** Eigenvector Centrality.  
**Middle Right ("D"):** Degree Centrality



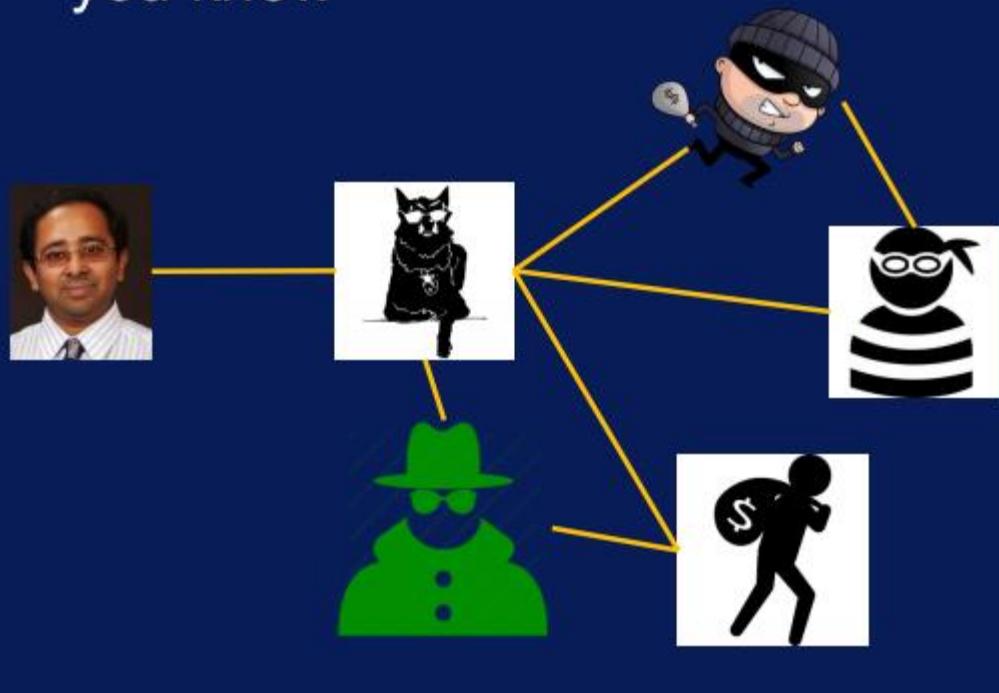
# Eigenvector centrality

- Centrality is proportional to the **sum of centrality of the neighbors**
- $C_i$  is proportional to  $\sum_{j:\text{friend of } i} C_j$
- $C_i = a \sum_j g_{ij} C_j$



# Eigenvector centrality: Interpretation

- “It’s not what you know but who you know”



# Pagerank

---

**Pagerank** is a variant of Eigenvector centrality

A random surfer of a graph is most likely to find nodes that are highly “central” because a lot of nodes point to these nodes directly or indirectly

# Comparison of different centrality measures

Id	Label	Grad	Closeness Centrality	Betweenness Centrality	Eigenvector Centrality
22	karev	9	0.38961	0.194009	1.0
38	sloan	7	0.410959	0.257873	0.716206
42	torres	5	0.416667	0.146802	0.697546
3	altman	4	0.333333	0.169435	0.294612
18	grey	4	0.283019	0.067774	0.249274
21	izzie	4	0.319149	0.071154	0.475488
32	o'malley	4	0.352941	0.082337	0.497948
1	addison	3	0.394737	0.080362	0.522082
8	bailey	3	1.0	0.003322	0.040869
12	chief	3	0.625	0.007752	0.053202
24	lexi	3	0.365854	0.057443	0.522432
44	yang	3	0.217391	0.063123	0.072553
7	avery	2	0.275229	0.003599	0.22715
15	derek	2	0.309278	0.028073	0.215354
16	ellis grey	2	0.625	0.006645	0.046777
23	kepner	2	0.288462	0.008079	0.327235
27	megan	2	1.0	0.001107	0.02113
31	olivia	2	0.306122	0.008031	0.400993
33	owen	2	0.265487	0.089701	0.119936
41	thatch grey	2	0.5	0.00443	0.035611

# Agenda

---

1. Introduction and examples
2. Fundamental concepts
  1. Creating graphs
  2. Visualizing graphs
  3. Centrality
  4. **Describing the macro structure**
  5. Community detection
3. Getting and handling data
4. Extended concepts
5. Discussion and conclusion

# Describing complex relationships

---

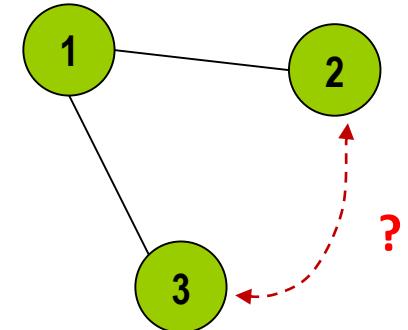
- **Global properties of the network (macro level)**
  - Degree distribution, average path length, density
- Segregation patterns (meso level)
  - Types of nodes, modularity, and homophily
- Local properties of single nodes (micro level)
  - Clustering coefficient, transitivity, ...
  - Position of a node: Centrality, prestige, robustness,...

# Clustering coefficient: from local to global

- ▶ How many of Grey's former lovers did have a romantic relationship with each other?
- ▶ **Local clustering coefficient:**

$$CI_i(g) = \frac{\#\{kj \text{ in } g | k, j \text{ in } N_i(g)\}}{\#\{kj | k, j \text{ in } N_i(g)\}}$$

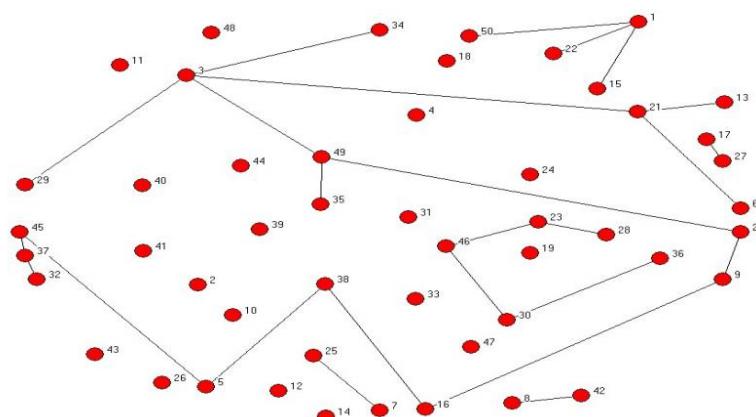
Frequency of this link?



- ▶ **Global clustering coefficient:**  
Scaled-up version averaged over all nodes in der network

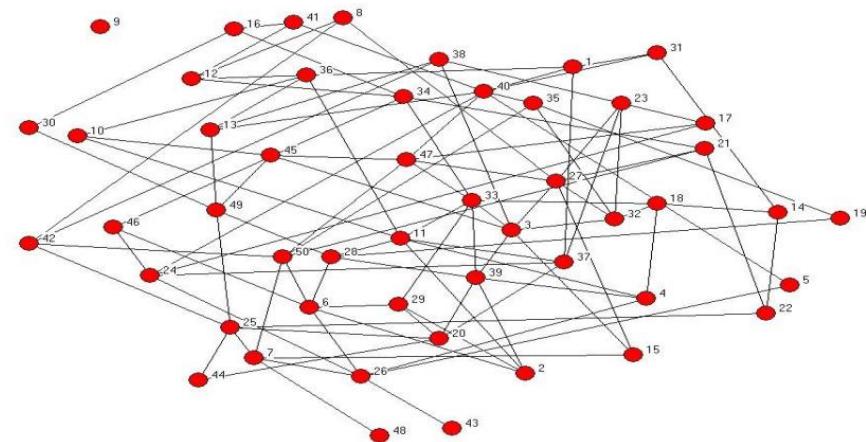
# Density

A **dense network** is a **network** in which the number of links of each node is close to the maximal number of nodes. Each node is linked to almost all other nodes. The total connected case in which exactly each node is linked to each other node is called a completely connected **network**. Value ranges between 0 and 1.



Random network  $p=0.02$ ,  $n=50$

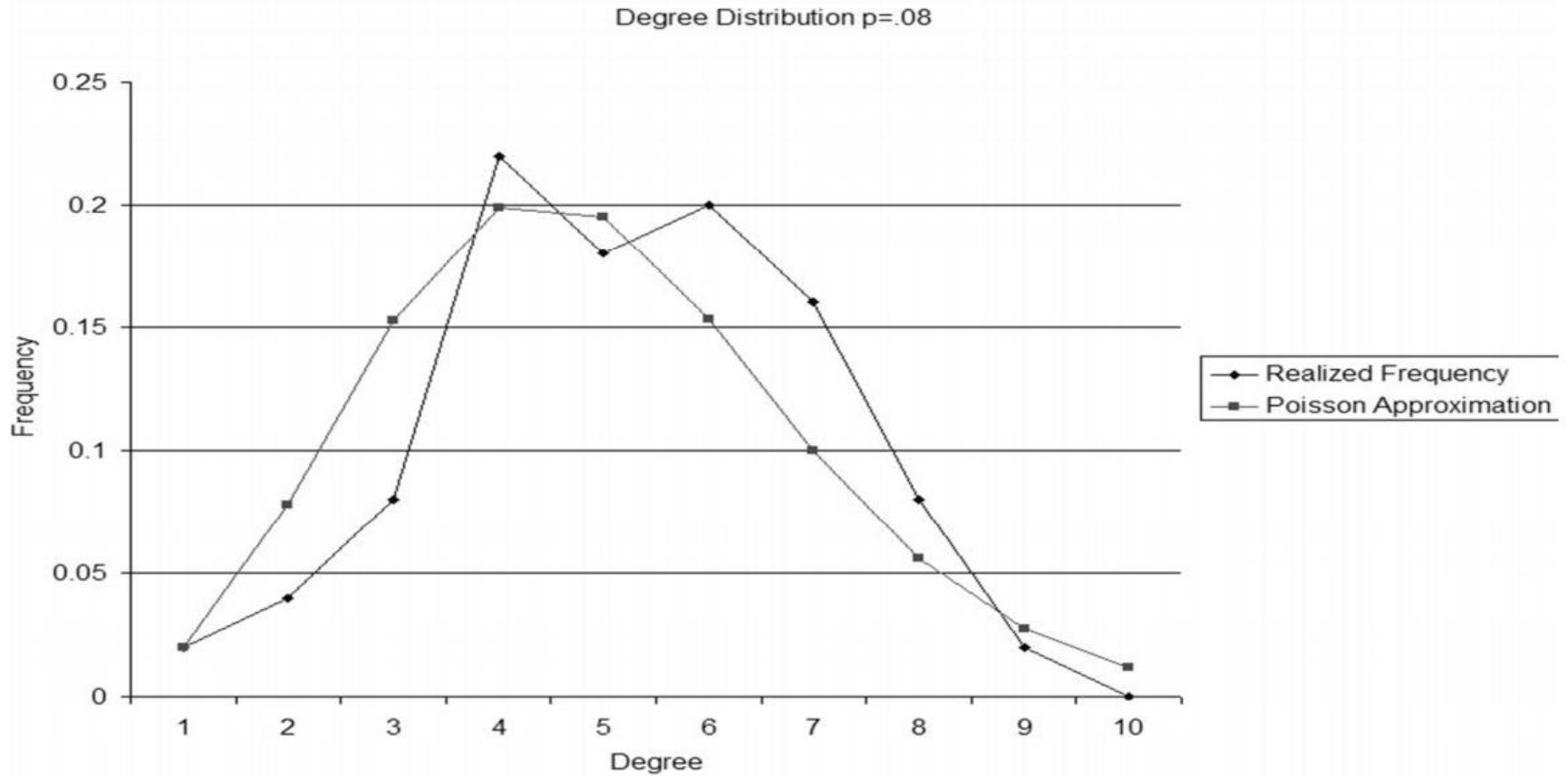
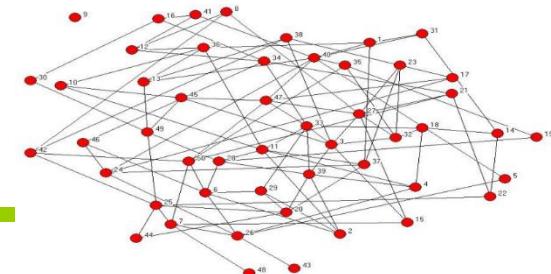
**Density: 0.019**



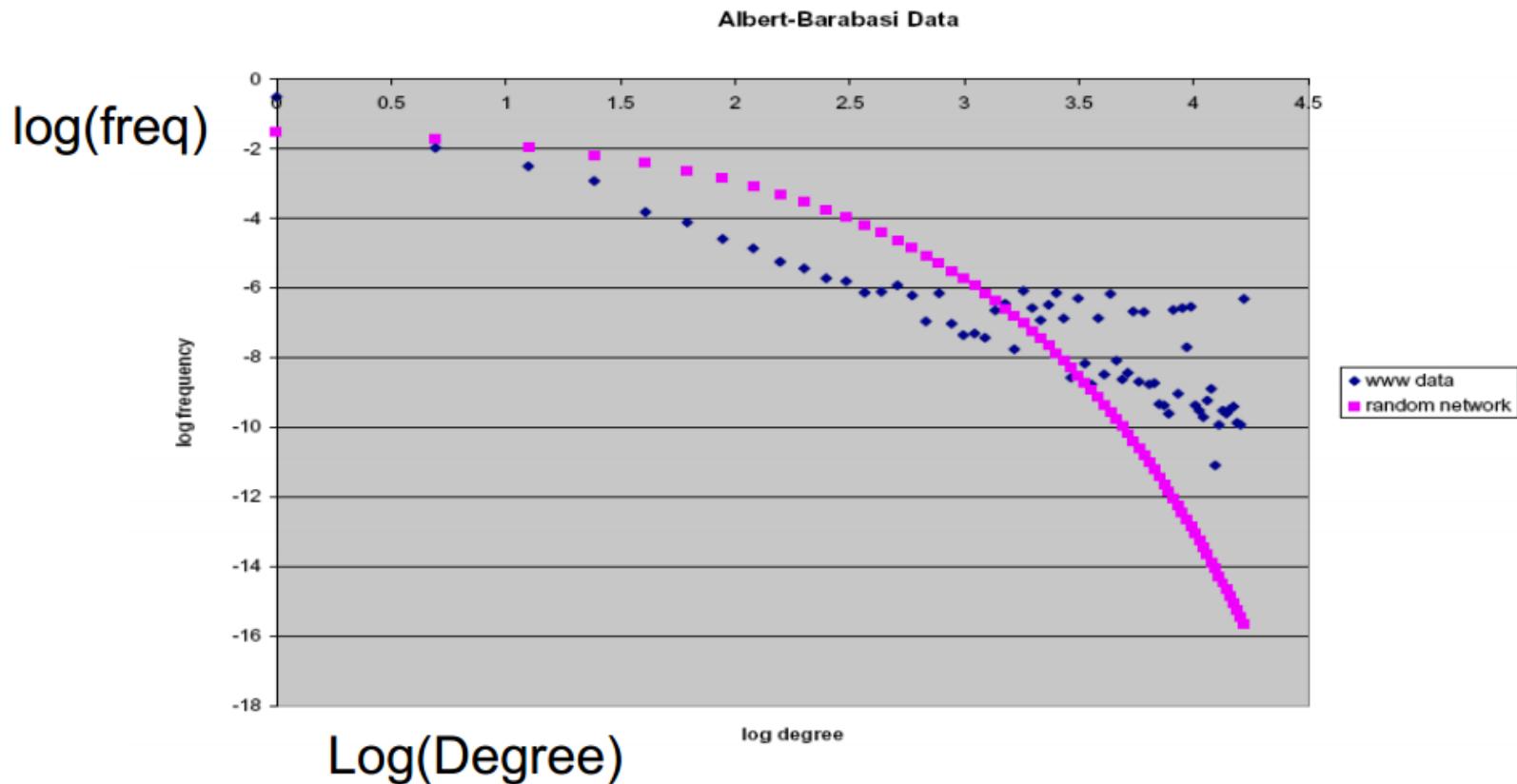
Random network  $p=0.05$ ,  $n=50$

**Density: 0.055**

# Degree distribution: Random network



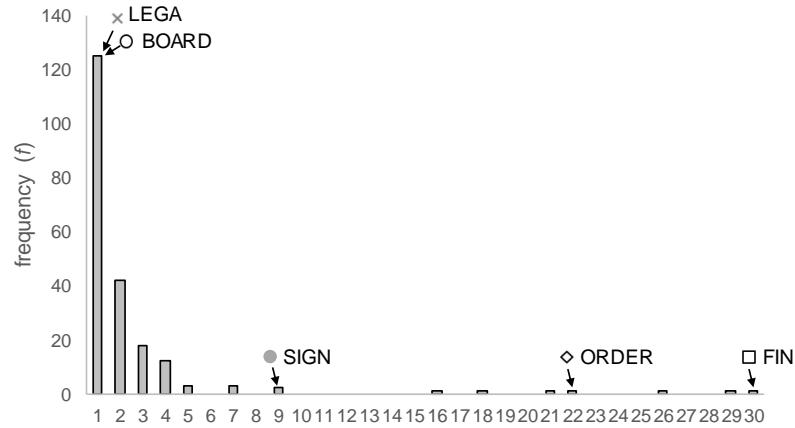
# Degree distribution: Scale-free network



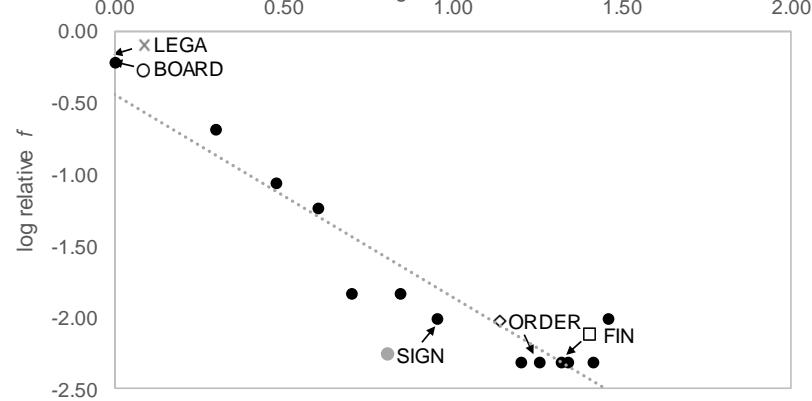
- “Fat Tails” (Price 1965), “Matthew Effect” (The rich get richer, the poor get poorer”) → More degrees with high or low degrees, middle range ill-populated (no reg. to mean)

# Degree distribution: Scale-free network

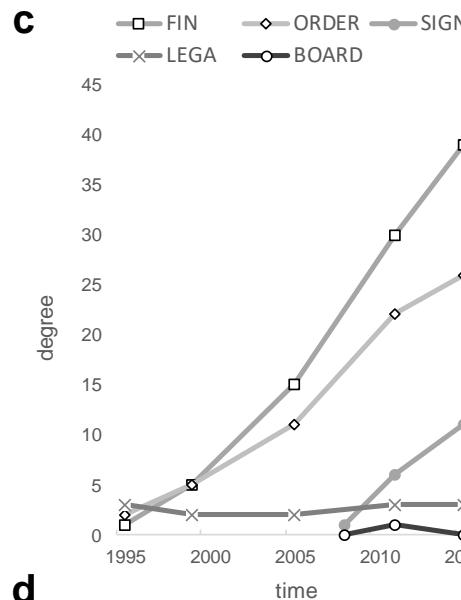
a



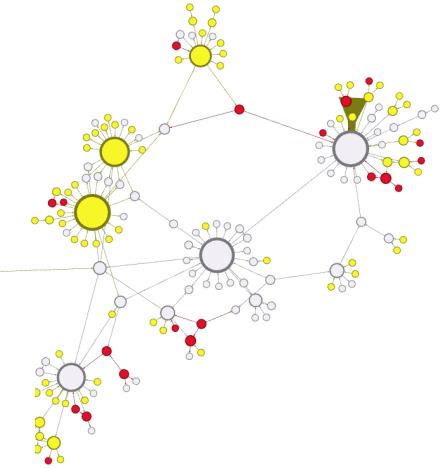
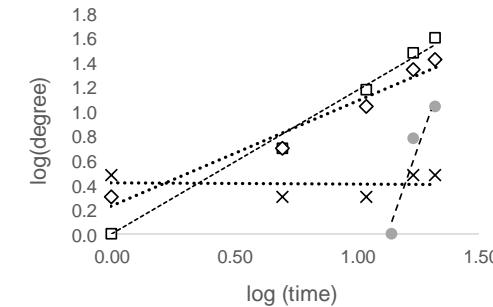
b



c



d



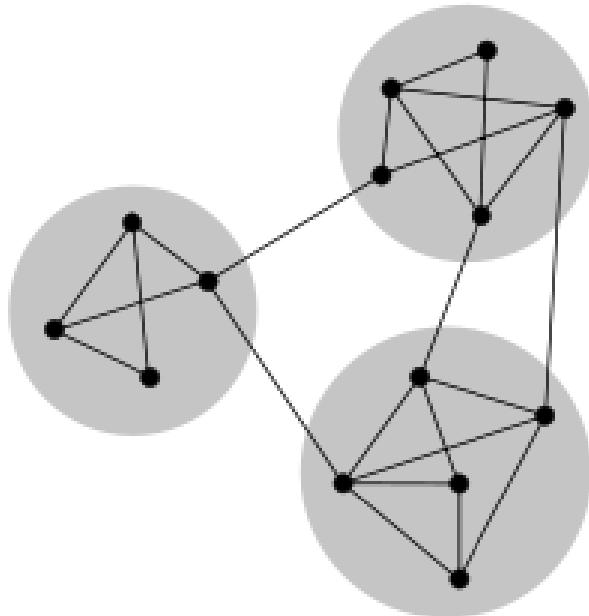
# Agenda

---

1. Introduction and examples
2. Fundamental concepts
  1. Creating graphs
  2. Visualizing graphs
  3. Centrality
  4. Describing the macro structure
  5. **Community detection**
3. Getting and handling data
4. Extended concepts
5. Discussion and conclusion

# Community detection: intuition

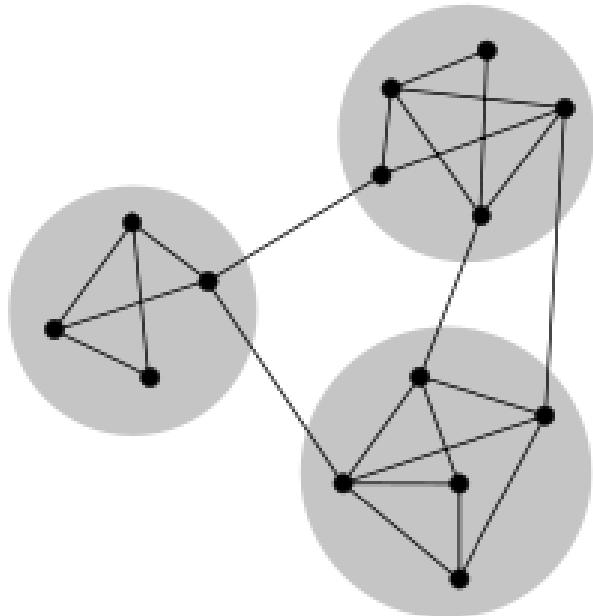
- ▶ “Where do I belong?”
  - ▶ Definitions
  - ▶ Modularity clustering
  - ▶ K-cores community detection



# Community detection: intuition

**Communities** (also called partitions, clusters, groupings) in networks are sets of nodes that are more densely connected within a community than between other communities.

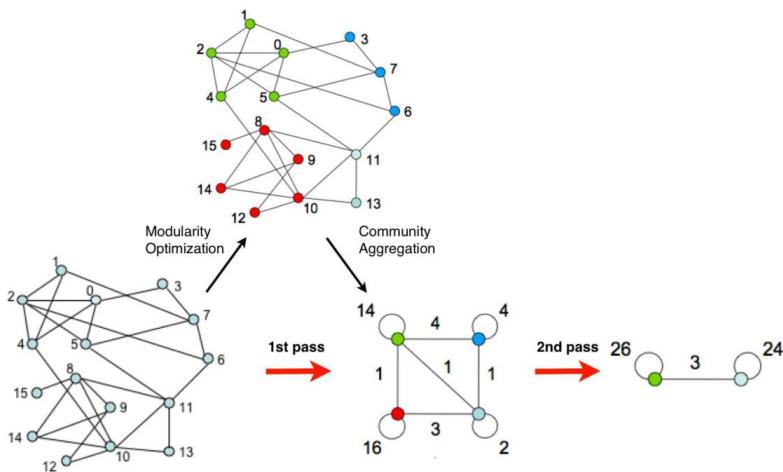
**Community detection** aims to reveal underlying community structure and can be handled in several different ways.



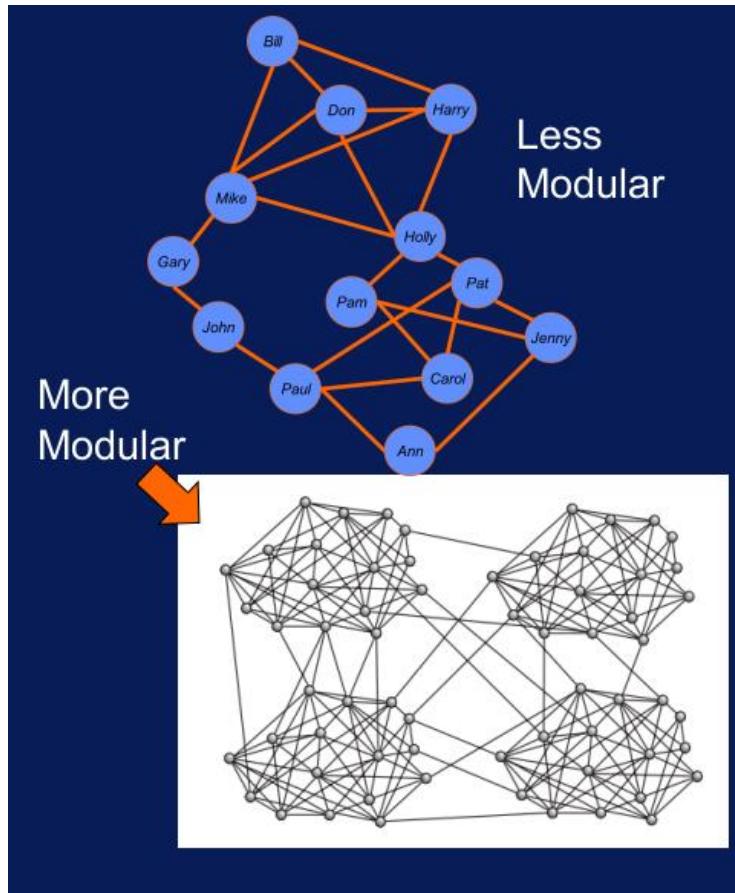
# Modularity clustering: intuition

## Modularity: A global measure of cluster quality

- fraction of the edges within the given groups minus the expected such fraction if edges were distributed at random.
- Value of the modularity lies in the range  $[-1/2, 1]$ .
- Steps of the method:

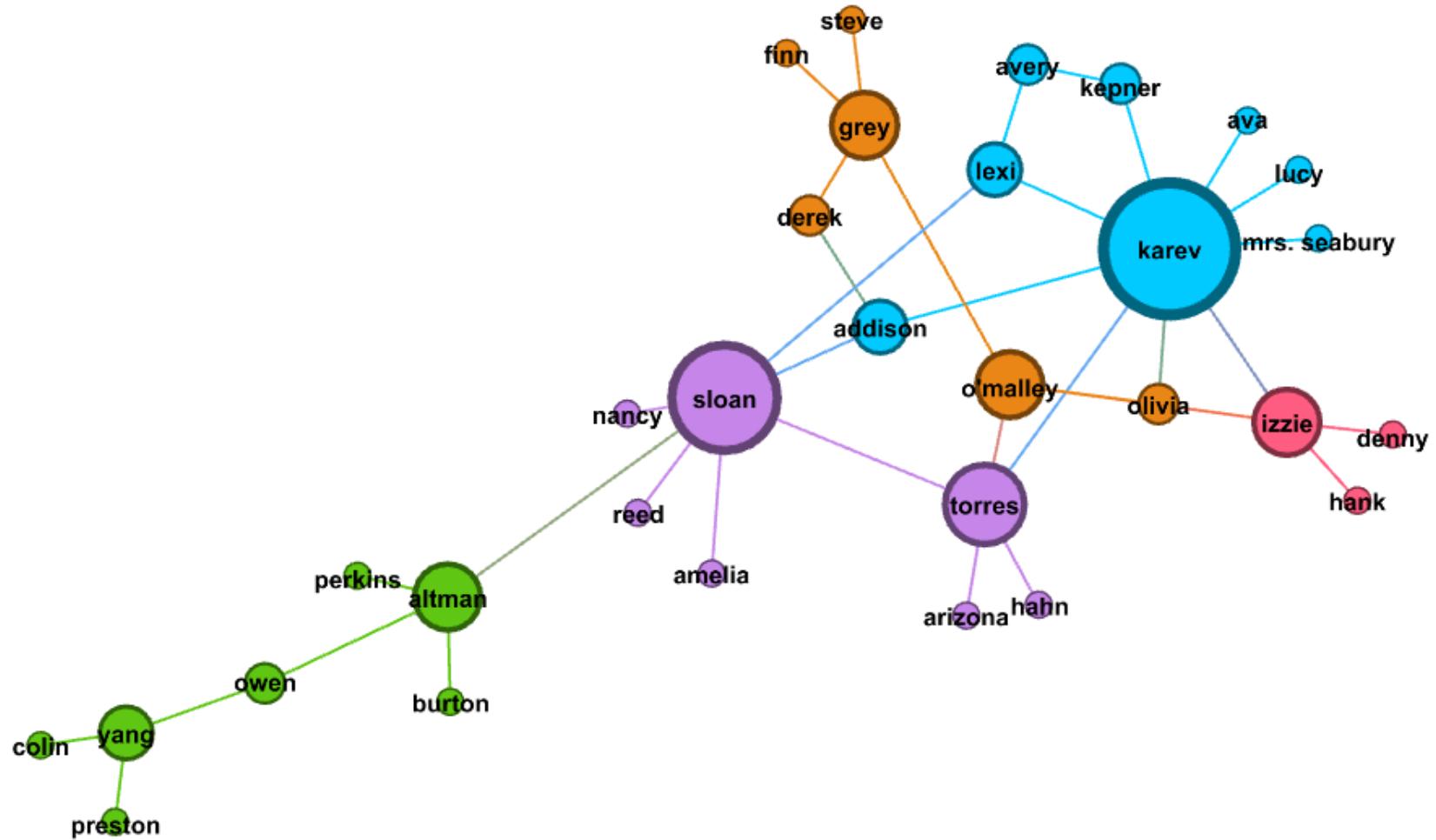


**Figure 1.** Visualization of the steps of our algorithm. Each pass is made of two phases: one where modularity is optimized by allowing only local changes of communities; one where the found communities are aggregated in order to build a new network of communities. The passes are repeated iteratively until no increase of modularity is possible.



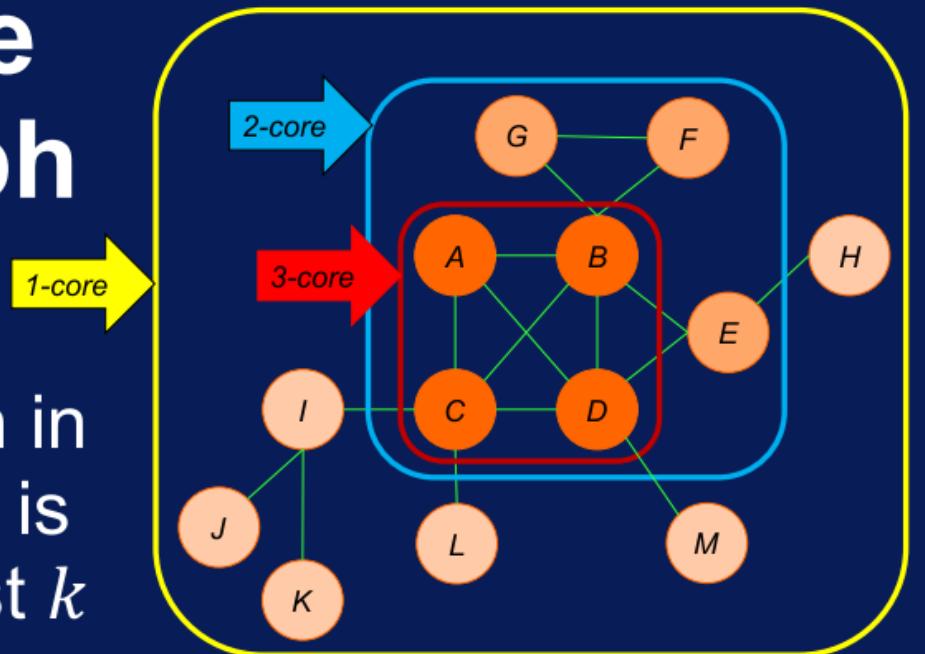
[Blondel et al. 2008]

# Modularity clustering: Different subgroups in the Grey's Anatomy cast



## Finding dense parts of a graph

- **$k$ -core**
  - Maximal subgraph in which each vertex is adjacent to at least  $k$  other vertices of the subgraph



# Agenda

---

1. Introduction and examples

2. Fundamental concepts

- 1. Creating graphs

- 2. Visualizing graphs

- 3. Centrality

- 4. Community detection

3. Getting and handling data

4. Extended concepts

5. Discussion and conclusion

# The “classical” workflow for an SNA study

---

- ▶ Formulate problem scope
- ▶ Build hypotheses
- ▶ Collect data
  - ▶ Survey data
  - ▶ Archival data
- ▶ Visualize data
- ▶ Analyze data

# General ways to extract data from social media sites and other web sources

## Ready-made software (e.g. Netvizz)

- Uses existing APIs such as FBs Graph API, Twitters Streaming API or GitHub's REST API
- Pros: Easy to use, intuitive
- Cons: Often limited in scope or functionality, changes in API may make the program obsolete

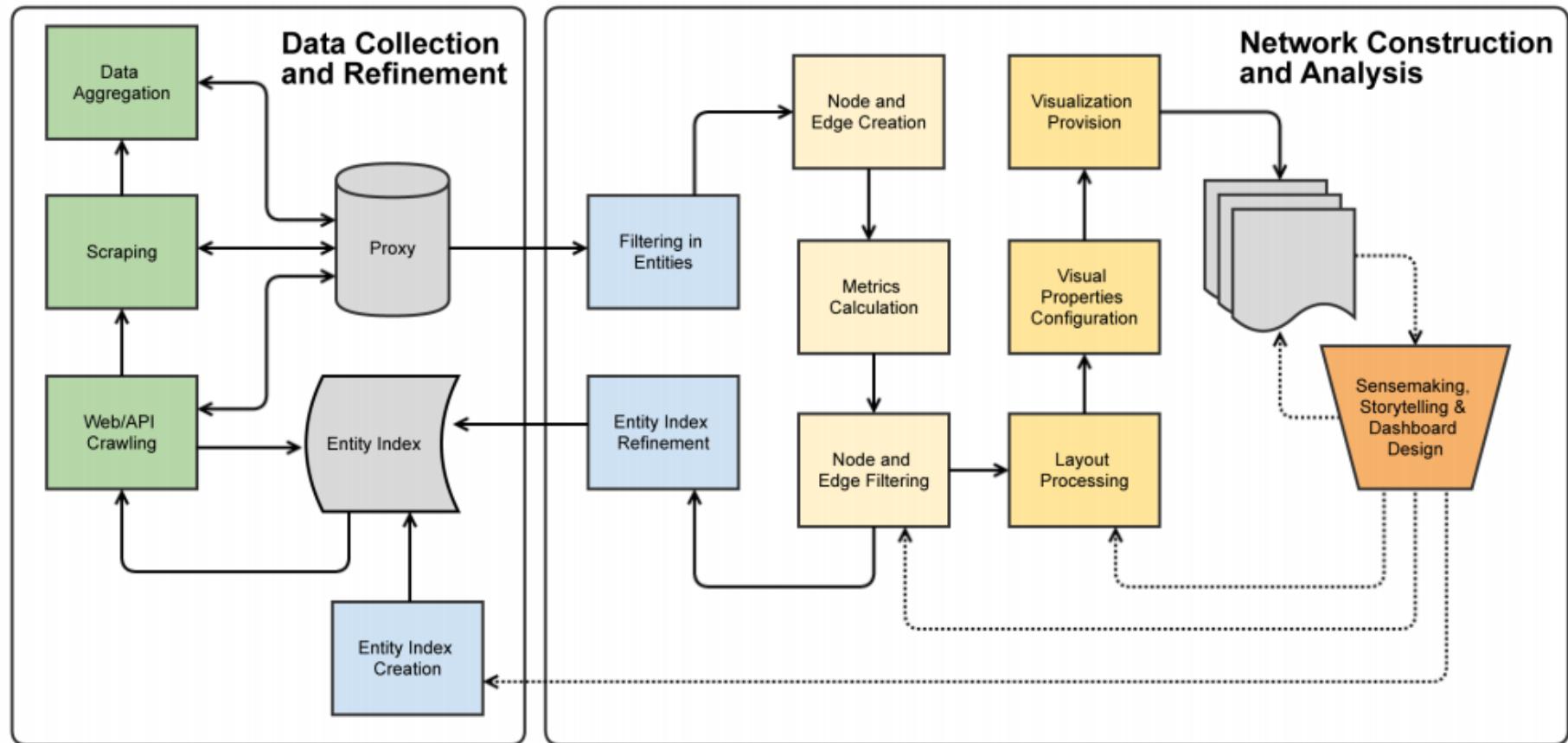
## Custom-developed code (e.g. Phyton)

- Similarly builds on existing APIs but uses self-developed program code (e.g. [Phyton](#) and [Project Jupyter](#))
- Pros: More flexible than ready-made software
- Cons: Learning curve, restricted by the APIs

## Web scraping (e.g. Web scraper Chrome extension)

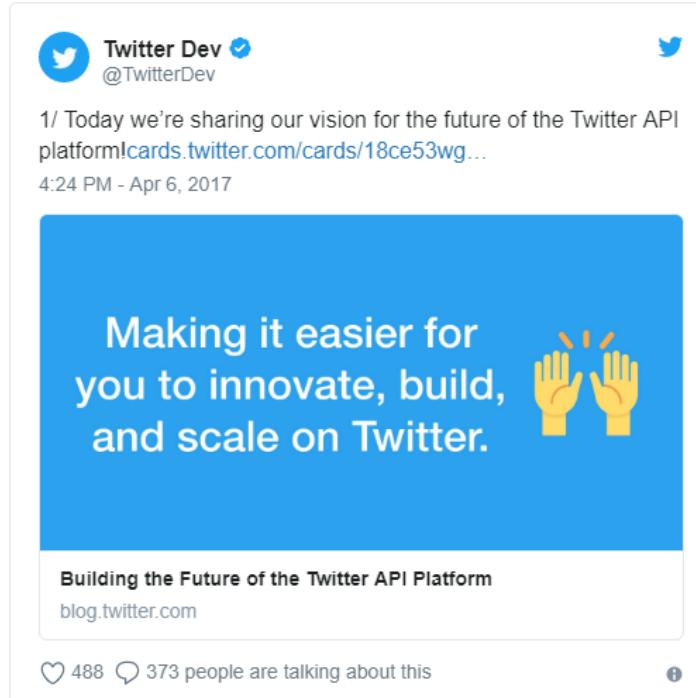
- Uses [scrapers](#) to extract data from website code
- Pros: Most flexible approach, in principle you can get all data that you can see or query on the web
- Cons: Often very messy websites “polluted” with Java Script, results may be mixed for various reasons

# The data science paradigm to an SNA study...



[Huhtamäki 2016]

# Twitter Streaming API: Understand the data structure



The JSON will be a mix of 'root-level' attributes (here we are highlighting some of the most fundamental attributes), and child objects (which are represented here with the `{}` notation):

```
{  
  "created_at": "Thu Apr 06 15:24:15 +0000 2017",  
  "id": 850006245121695744,  
  "id_str": "850006245121695744",  
  "text": "1/ Today we're sharing our vision for the future of the Twitter API platform!nhttps://t.co/XweGngmxIP",  
  "user": {},  
  "entities": {}  
}
```

# Twitter Streaming API: Possible networks to create

- ▶ re-tweeting (retweet network)
- ▶ replying (reply network) to existing tweets
- ▶ mentioning (mention network) other users
- ▶ friends/followers social relationships among user involved in the above activities

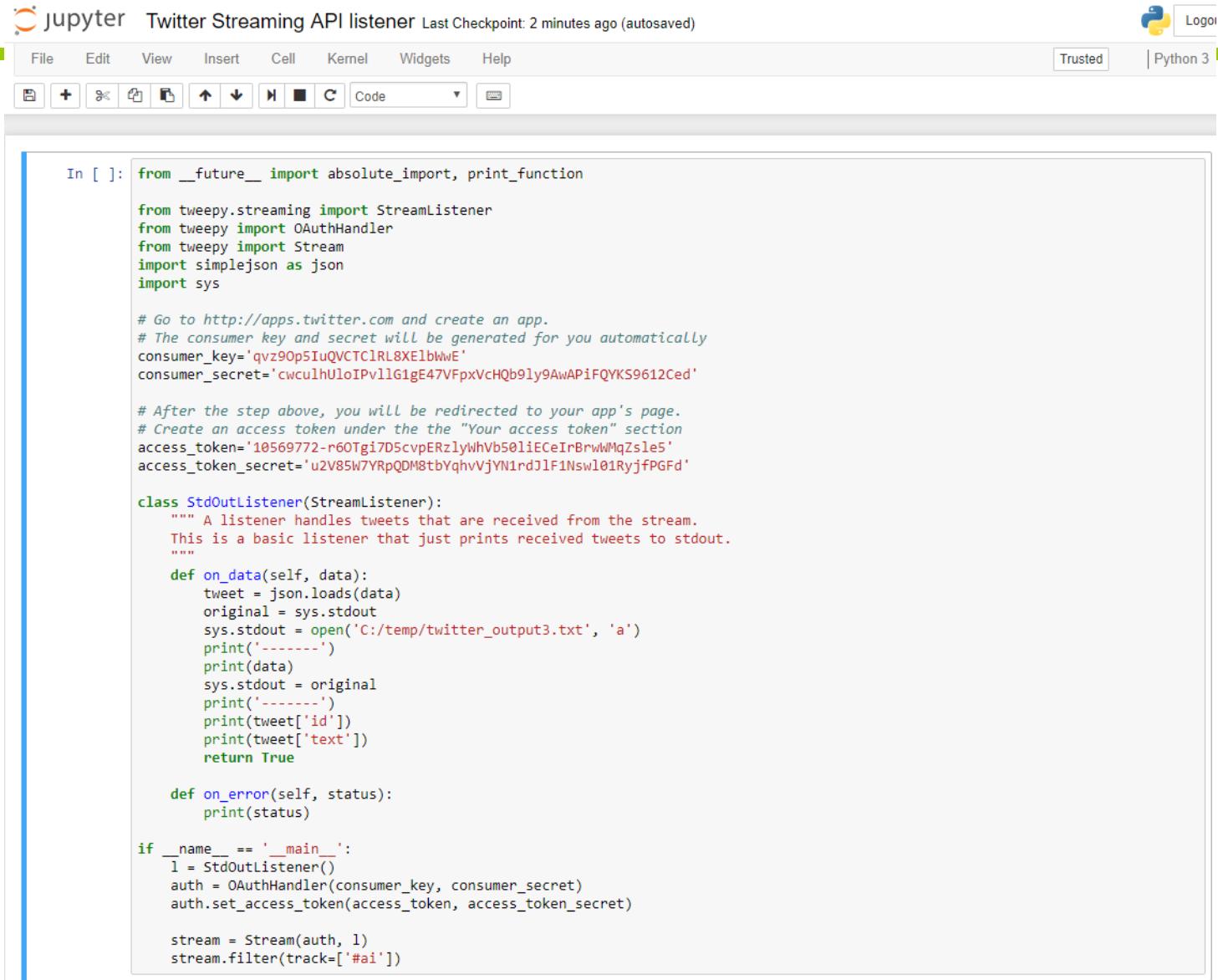
[<http://snap.stanford.edu/data/higgs-twitter.html>]

# Twitter Streaming API: Getting set-up

1. Download Anaconda (includes Phyton)
2. Download Project Jupyter Notebook
3. Install the necessary packages (tweepy, simplejson)
4. Create a Twitter account
5. Create a Twitter application
6. Create the listener (next slide)

[<https://github.com/jukkahuhtamaki/demo-twitter-collector>]

# Twitter API: Building a listener with jupyter and tweepy



The screenshot shows a Jupyter Notebook interface with the title "Twitter Streaming API listener". The notebook has a single cell (In [ ]:) containing Python code. The code imports necessary modules from tweepy and defines a StdOutListener class that implements the StreamListener interface. It handles tweet data by printing it to stdout and saving it to a file named 'twitter\_output3.txt'. The code also includes consumer key and secret values, and a stream configuration to track the '#ai' hashtag.

```
In [ ]: from __future__ import absolute_import, print_function

from tweepy.streaming import StreamListener
from tweepy import OAuthHandler
from tweepy import Stream
import simplejson as json
import sys

# Go to http://apps.twitter.com and create an app.
# The consumer key and secret will be generated for you automatically
consumer_key='qvz90p5IuQVCTC1RL8XElbwE'
consumer_secret='cwcuhluUoIPvllGigE47VFpxVcHQb9ly9AwAPiFQYKS9612Ced'

# After the step above, you will be redirected to your app's page.
# Create an access token under the the "Your access token" section
access_token='10569772-r6OTg17D5cvpErzlyWhvb50liEcIrBrwMMqzsle5'
access_token_secret='u2V85W7YRpQDM8tbYqhVjYN1rdJlF1Nswl01RyjfPGFd'

class StdOutListener(StreamListener):
    """ A listener handles tweets that are received from the stream.
    This is a basic listener that just prints received tweets to stdout.
    """
    def on_data(self, data):
        tweet = json.loads(data)
        original = sys.stdout
        sys.stdout = open('C:/temp/twitter_output3.txt', 'a')
        print('-----')
        print(data)
        sys.stdout = original
        print('-----')
        print(tweet['id'])
        print(tweet['text'])
        return True

    def on_error(self, status):
        print(status)

if __name__ == '__main__':
    l = StdOutListener()
    auth = OAuthHandler(consumer_key, consumer_secret)
    auth.set_access_token(access_token, access_token_secret)

    stream = Stream(auth, l)
    stream.filter(track=['#ai'])
```

# Twitter API: Writing output to stdout

```
cmd: Eingabeaufforderung - start_demo
RT @ipfconline1: Three Barriers to #AI Adoption Across The Enterprise
https://t.co/tv3bSbMLzq v/ @theterminal
#DigitalTransformation

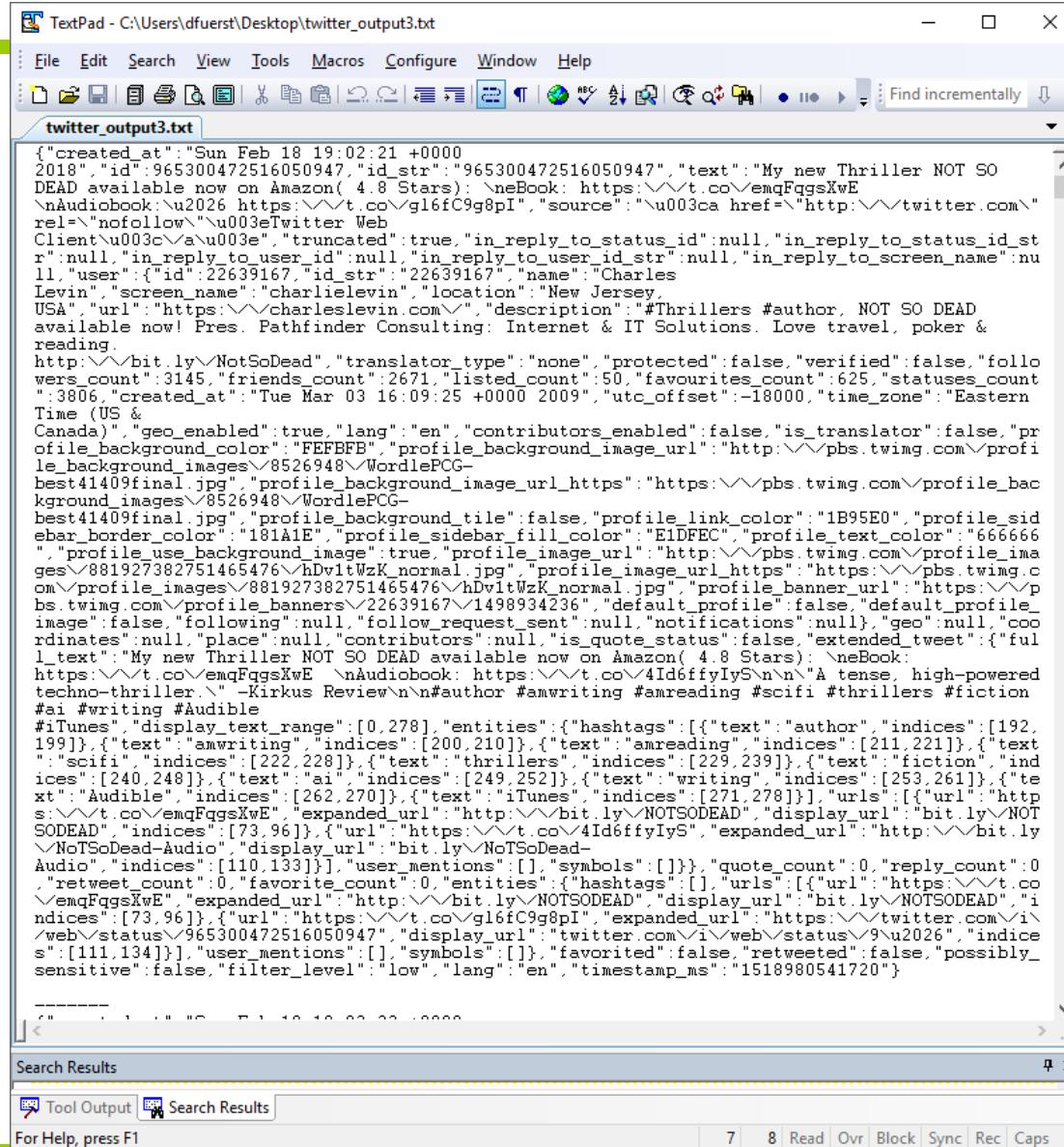
Cc @...
-----
965343189392650241
RT @ipfconline1: Three Barriers to #AI Adoption Across The Enterprise
https://t.co/tv3bSbMLzq v/ @theterminal
#DigitalTransformation

Cc @...
-----
965343194576834561
RT @Ronald_vanLoon: A Warrior #Robot that can jump 4 feet in the Air
via @mashable |

#Robotics #AI #ArtificialIntelligence #MachineLearni...
-----
965343200343928843
#RT @MktgSciences: Connecting firms with talented IoT professionals ==> https://t.co/HOTP7s05Vr
#IoT #IIoT.. https://t.co/s0B1w0CYmI
-----
965343202751516672
#LIKE PLZ RT IT :)

7 Building Blocks for Success in the #InternetofThings
https://t.co/xneYHOneAg
#IoT #AI.. https://t.co/QyBowfRQ0x
-----
965343212914315264
RT @emaware: https://t.co/gnRglc0hq1 We are hiring! Answer 6 questions and join an amazing team who thrive to empower users with emotional...
-----
965343218333253633
#RT @MktgSciences: Google's #AI Guru @demishassabis Says That Great Artificial Intelligence Must Build On Neuroscie.. https://t.co/F8cckGzMgK
```

# Twitter API: Writing output to file



The screenshot shows a window titled "TextPad - C:\Users\dfuerst\Desktop\twitter\_output3.txt". The window contains a large amount of JSON-formatted Twitter API output. The output describes a tweet from a user named "Charles Levin" (@charlielevin) with ID 22639167. The tweet was created at "Sun Feb 18 19:02:21 +0000 2018" and has a long URL: <https://bit.ly/NotSoDead>. The tweet content is: "My new Thriller NOT SO DEAD available now on Amazon( 4.8 Stars): \nneBook: https://t.co/emeqFqgsXwE \nAudioobook:\u2026 https://t.co/g16fC9g8pI", followed by a link to the tweet's page on Twitter. The tweet has 3145 followers and 2671 friends. It includes profile and background images from WordlePCG. The tweet is in English and has a timestamp of "Tue Mar 03 16:09:25 +0000 2009". The JSON structure also includes entities like hashtags and URLs.

# Twitter API: Parsing the file to create data set I

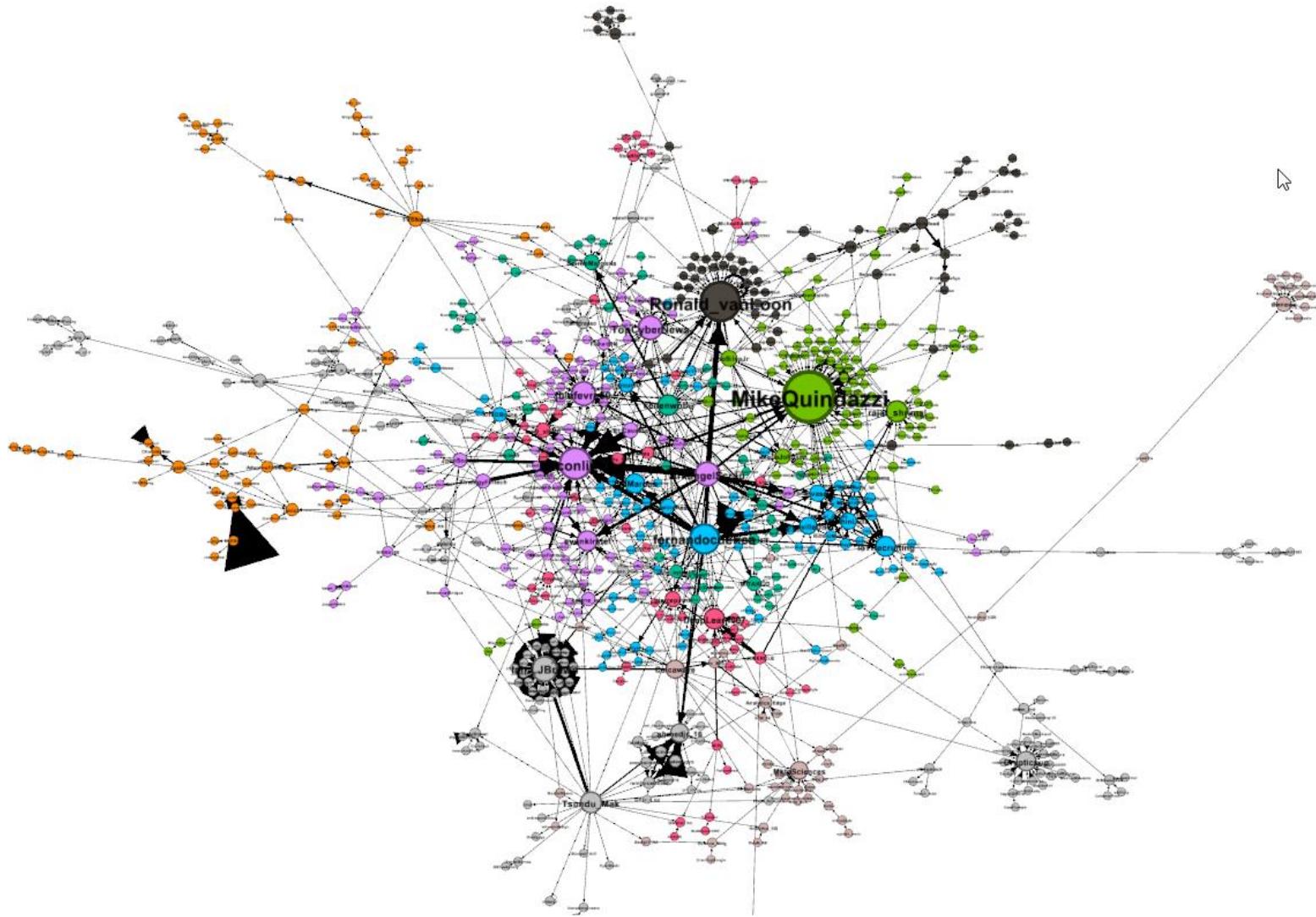
A	B	C
Created_at	Tweet_text	Screen_name
1		RT
2	Created_at	Screen_name
3	Sun Feb 18 19:02:23 +0000 2l RT @AbyssCreations: Oh you know, just casual office attire. #abysscreations #realdoll #realdolls #harmonyAI #AI #artif LordRiehl	AbyssCreations:
4	Sun Feb 18 19:02:23 +0000 2l RT @guidautonoma: Where is #ElonMusk's #TeslaRoadster with #Starman? In There is a site... In #SpaceX #Tesla # JulioSilvaJr	guidautonoma:
5	Sun Feb 18 19:02:23 +0000 2l RT @MarexSolutions: Are you a #CFO? Your #career is no doubt undergoing masses of change with the evolution of # WIOMAX_MD	MarexSolutions:
10	Sun Feb 18 19:02:25 +0000 2l RT @AbyssCreations: MF#Youtu2019re part of the worldu2019s greatest #lovedoll company, but @Twitter won't! LordRiehl	AbyssCreations:
11	Sun Feb 18 19:02:25 +0000 2l RT @SpirosMargaris: #Chinamu2019s #tech industry Innis catching up with #SiliconValley Invia @TheEconomist InInhi Massoud	SpirosMargaris:
14	Sun Feb 18 19:02:28 +0000 2l RT @reinforceabtwt: 7 Best Analytics & Automation Courses Online For #Business InInhttps://Wt.co/L6TnJis9 Jadirectivestwt	reinforceabtwt:
15	Sun Feb 18 19:02:30 +0000 2l RT @singularity_net: ICYDK: AI is whatever hasn't been done yet." - Douglas Hofstadter InSunday #Inspiration #AI # 000Sergenio	singularity_net:
16	Sun Feb 18 19:02:31 +0000 2C RT @lainLJBrown: Slowly, but surely, #AI use is rising across industries https://Wt.co/YemgTMGdWQ https://Wt.co/lqb4 FabienBrodie	lainLJBrown:
19	Sun Feb 18 19:02:36 +0000 2l RT @iMariaJohnson: How #AI and #MachineLearning Will Impact Content in #SEO Inud83eudd13ud83dudc46https://Wt.co/JulioSilvaJr	iMariaJohnson:
20	Sun Feb 18 19:02:36 +0000 2l RT @ahmedjr_16: #Review of #MachineLearning A-Zlu2122 Hands-On #Python #R In #DataScience InInhttps://Wt.co/Jadirectivestwt	ahmedjr_16:
21	Sun Feb 18 19:02:37 +0000 2l RT @ahmedjr_16: #Review of #MachineLearning A-Zlu2122 Hands-On #Python #R In #DataScience InInhttps://Wt.co/fernandocuenca	ahmedjr_16:
23	Sun Feb 18 19:02:39 +0000 2l RT @singularity_net: ICYDK: AI is whatever hasn't been done yet." - Douglas Hofstadter InSunday #Inspiration #AI # nazkong	singularity_net:
24	Sun Feb 18 19:02:39 +0000 2l RT @MikeQuindazzi: 1,000 robots change the game on a traditional pick, pack, and ship warehouse reports the @w@ JulioSilvaJr	MikeQuindazzi:
25	Sun Feb 18 19:02:40 +0000 2l RT @iMariaJohnson: Build #Intelligent_chatbots ud83eudd13ud83dudc46In#chatbot #facebook #messengerbc JulioSilvaJr	iMariaJohnson:
27	Sun Feb 18 19:02:47 +0000 2l RT @raja_shrimal: This autonomous selfie drone has 13 camerasIn#Robotics #AI #ArtificialIntelligence #Drones #Te JulioSilvaJr	raja_shrimal:
28	Sun Feb 18 19:02:52 +0000 2l RT @mywoscotland: Is it really that hard to imagine where machines can outperform us in most workplaces? Via @Grc BH3pathways	mywoscotland:
30	Sun Feb 18 19:02:57 +0000 2l RT @emaware: https://Wt.co/IgnRgICoh1 We are hiring! Answer 6 questions and join an amazing team who thrive to chan_zagorskas	emaware:
31	Sun Feb 18 19:02:58 +0000 2l RT @SpirosMargaris: Human #speech will be replaced InInby #thought #communication by 2050, claims expert InIn JulioSilvaJr	SpirosMargaris:
35	Sun Feb 18 19:03:01 +0000 2C RT @ahmedjr_16: 17 Best #IT #Training & #Certification CoursesInInhttps://Wt.co/TJ6MNCMYlw InIn#AI #Analytic Jadirectivestwt	ahmedjr_16:
37	Sun Feb 18 19:03:02 +0000 2l RT @Ronald_vanLoon: Create #3D Masterpieces in #VR using BlocksInvia @blefevre60 InIn#VirtualReality #AI #artif JulioSilvaJr	Ronald_vanLoon:
39	Sun Feb 18 19:03:06 +0000 2l RT @ahmedjr_16: #Review of #DeepLearning A-Zlu2122 Hands-On Artificial #NeuralNetworksInInhttps://Wt.co/leto1W Jadirectivestwt	ahmedjr_16:
41	Sun Feb 18 19:03:09 +0000 2l RT @Ronald_vanLoon: Create #3D Masterpieces in #VR using BlocksInvia @blefevre60 InIn#VirtualReality #AI #artif Mada_GD	Ronald_vanLoon:
43	Sun Feb 18 19:03:11 +0000 20 RT @ahmedjr_16: 8 Best SQL Courses Online To Advance Your #CareerInInhttps://Wt.co/IloRevkRT11 InIn#AI #DataSc Jadirectivestwt	ahmedjr_16:
44	Sun Feb 18 19:03:11 +0000 20 RT @MktgSciences: Bankinglu2019s Strategic Roadmap: Priorities and Investments for The Year AheadIn2026 https://Wt.co/AL_Mentors	MktgSciences:
45	Sun Feb 18 19:03:11 +0000 20 RT @ahmedjr_16: 8 Best SQL Courses Online To Advance Your #CareerInInhttps://Wt.co/IloRevkRT11 InIn#AI #DataSc fernandocuenca	ahmedjr_16:
46	Sun Feb 18 19:03:13 +0000 2C RT @mjcavaretta: The best part of this article is at the end. If you can't scale your analytics practice nothing else matter JulioSilvaJr	mjcavaretta:
47	Sun Feb 18 19:03:15 +0000 2C RT @ahmedjr_16: 21 Best #DataScience Courses to Become a #DataScientist InInhttps://Wt.co/Mq4EYLZc InIn#AI Jadirectivestwt	ahmedjr_16:
49	Sun Feb 18 19:03:20 +0000 2l RT @MikeQuindazzi: 4 levels of #ArtificialIntelligence via #PwCin#Automated - tasks done faster, less-expensiveIn#A: Mada_GD	MikeQuindazzi:
50	Sun Feb 18 19:03:21 +0000 2C RT @TheMegaTrends: #Top10 Hot Data #Security And #Privacy TechnologiesInIn@Forbes @GillPressInIn#CyberSe JulioSilvaJr	TheMegaTrends:
53	Sun Feb 18 19:03:24 +0000 2l RT @AitheonOfficial: Andrew and Ryan photoed with @SebastianThrun, founder of Udacity and renowned for his cont 48Dman369	AitheonOfficial:
54	Sun Feb 18 19:03:24 +0000 2l RT @TopCyberNews: #Top10 Hot Data #Security And #Privacy TechnologiesInIn@Forbes @GillPressInIn#CyberSec JulioSilvaJr	TopCyberNews:
55	Sun Feb 18 19:03:24 +0000 2l RT @andi_staub: Good Read!InThe Most Prevalent Myths: What Are the Consequences of Confusion About #AI?In# Mada_GD	andi_staub:
57	Sun Feb 18 19:03:25 +0000 2l RT @andi_staub: Good Read!InThe Most Prevalent Myths: What Are the Consequences of Confusion About #AI?In# fernandocuenca	andi_staub:
58	Sun Feb 18 19:03:25 +0000 2l RT @devnullius: Did you have adaptive videos when you went to school? Customized to your personal likings, styles & itknowingness	devnullius:
59	Sun Feb 18 19:03:26 +0000 2l RT @ProductiveSys: How #Blockchain and #SmartContracts Will Change Contract Management in 2018 - @icertis In# sajidmizra	ProductiveSys:
61	Sun Feb 18 19:03:28 +0000 2l RT @MikeQuindazzi: Next in #gitreach? #Robotics researchers are pioneering #robots for #farming and #driverless trac JulioSilvaJr	MikeQuindazzi:
62	Sun Feb 18 19:03:28 +0000 2l RT @appliedAlbook: All the matrix #calculus you need for #deeplearning has been conveniently packaged & clar ryanbriggs	appliedAlbook:
63	Sun Feb 18 19:03:29 +0000 2l RT @Gartner_inc: Discover, What to Do and Not Do With #AI, in this webinar: https://Wt.co/zE2XVRK33 @WhitAndrew mskohli	Gartner_inc:
69	Sun Feb 18 19:03:31 +0000 2C RT @singularity_net: ICYDK: AI is whatever hasn't been done yet." - Douglas Hofstadter InSunday #Inspiration #AI # DNBR5117	singularity_net:
70	Sun Feb 18 19:03:32 +0000 2l RT @AL_Mentors: #RT @MktgSciences: Bankinglu2019s Strategic Roadmap: Priorities and Investments for The Year A Calcaware	AL_Mentors:
71	Sun Feb 18 19:03:32 +0000 2l RT @ahmedjr_16: 8 Best SQL Courses Online To Advance Your #CareerInInhttps://Wt.co/IloRevkRT11 InIn#AI #DataSc calcaware	ahmedjr_16:
73	Sun Feb 18 19:03:35 +0000 2l RT @chboursin: The 11 clusters of InnovationInIn#IoT #Entrepreneur #AI #tech #SMM #Fintech #startup #MachineLea mik072	chboursin:
74	Sun Feb 18 19:03:36 +0000 2l RT @chboursin: The 11 clusters of InnovationInIn#IoT #Entrepreneur #AI #tech #SMM #Fintech #startup #MachineLea fernandocuenca	chboursin:
76	Sun Feb 18 19:03:36 +0000 2l RT @AbyssCreations: Harmony's ready for her closeup, she loves getting in front of the camera. #harmonyAI #abysscri LordRiehl	AbyssCreations:
77	Sun Feb 18 19:03:37 +0000 2l RT @SaleMove: The biggest misconception about #artificialintelligence. #AhInhttps://Wt.co/DhSwk1THye https://Wt.cc/JulioSilvaJr	SaleMove:
78	Sun Feb 18 19:03:38 +0000 2l RT @ProductiveSys: #ArtificialIntelligence can help you protect your personal #data In#AI #ML #DL #Algorithms #NLP sajidmizra	ProductiveSys:
79	Sun Feb 18 19:03:38 +0000 2l RT @ProductiveSys: #ArtificialIntelligence can help you protect your personal #data In#AI #ML #DL #Algorithms #NLP fernandocuenca	ProductiveSys:
80	Sun Feb 18 19:03:38 +0000 2l RT @ipfonline1: #MachineLearning Translation and the Google Translate AlgorithmInInhttps://Wt.co/vubSXY2H60 v Mada_GD	ipfonline1:
81	Sun Feb 18 19:03:39 +0000 2l RT @bonzailegioners: The evolution of #MachineLearning 2010 to 2040: https://Wt.co/o2C0yeel4T via @MikeQuindz hiroyasak	bonzailegioners:
82	Sun Feb 18 19:03:40 +0000 2l RT @terminus7ai: What #PredictiveAnalytics, #BigData And The Rise Of #ArtificialIntelligence Mean For Real Estate ht JulioSilvaJr	terminus7ai:
84	Sun Feb 18 19:03:42 +0000 2l RT @ileadgen: 7 Best #MobileApps #Development Courses for Beginners InInhttps://Wt.co/Y1prA7vZ0 InIn#AppDe Jadirectivestwt	ileadgen:



# Twitter API: Parsing the file to create data set II

	A	B	C
1	Source	Target	Type
2	LordRiehl	AbyssCreations	Directed
3	JulioSilvaJr	guidautonoma	Directed
4	WIOMAX_MD	MarexSolutions	Directed
5	LordRiehl	AbyssCreations	Directed
6	Masssoud	SpirosMargaris	Directed
7	Jadirectivestwt	reinforceLabwt	Directed
8	OOOSergenio	singularity_net	Directed
9	FabienBrodie	IainLJBrown	Directed
10	JulioSilvaJr	iMariaJohnsen	Directed
11	Jadirectivestwt	ahmedjr_16	Directed
12	fernandocuenca	ahmedjr_16	Directed
13	nazkong	singularity_net	Directed
14	JulioSilvaJr	MikeQuindazzi	Directed
15	JulioSilvaJr	iMariaJohnsen	Directed
16	JulioSilvaJr	rajat_shrimal	Directed
17	BHSfpathways	mywowscotland	Directed
18	chan_zagorskas	emaware	Directed
19	JulioSilvaJr	SpirosMargaris	Directed
20	Jadirectivestwt	ahmedjr_16	Directed
21	JulioSilvaJr	Ronald_vanLoon	Directed
22	Jadirectivestwt	ahmedjr_16	Directed
23	Mada_GD	Ronald_vanLoon	Directed
24	Jadirectivestwt	ahmedjr_16	Directed
25	AI_Mentors	MktgSciences	Directed
26	fernandocuenca	ahmedjr_16	Directed
27	JulioSilvaJr	mjcavaretta	Directed
28	Jadirectivestwt	ahmedjr_16	Directed
29	Mada_GD	MikeQuindazzi	Directed
30	JulioSilvaJr	TheMegaTrends	Directed
31	48Diman369	AitheonOfficial	Directed
32	JulioSilvaJr	TopCyberNews	Directed
33	Mada_GD	andi_staub	Directed
34	fernandocuenca	andi_staub	Directed
35	itknowingness	devnullius	Directed
36	sajidmirza	ProductiveSys	Directed
37	JulioSilvaJr	MikeQuindazzi	Directed

# Import to Gephi for vizualization

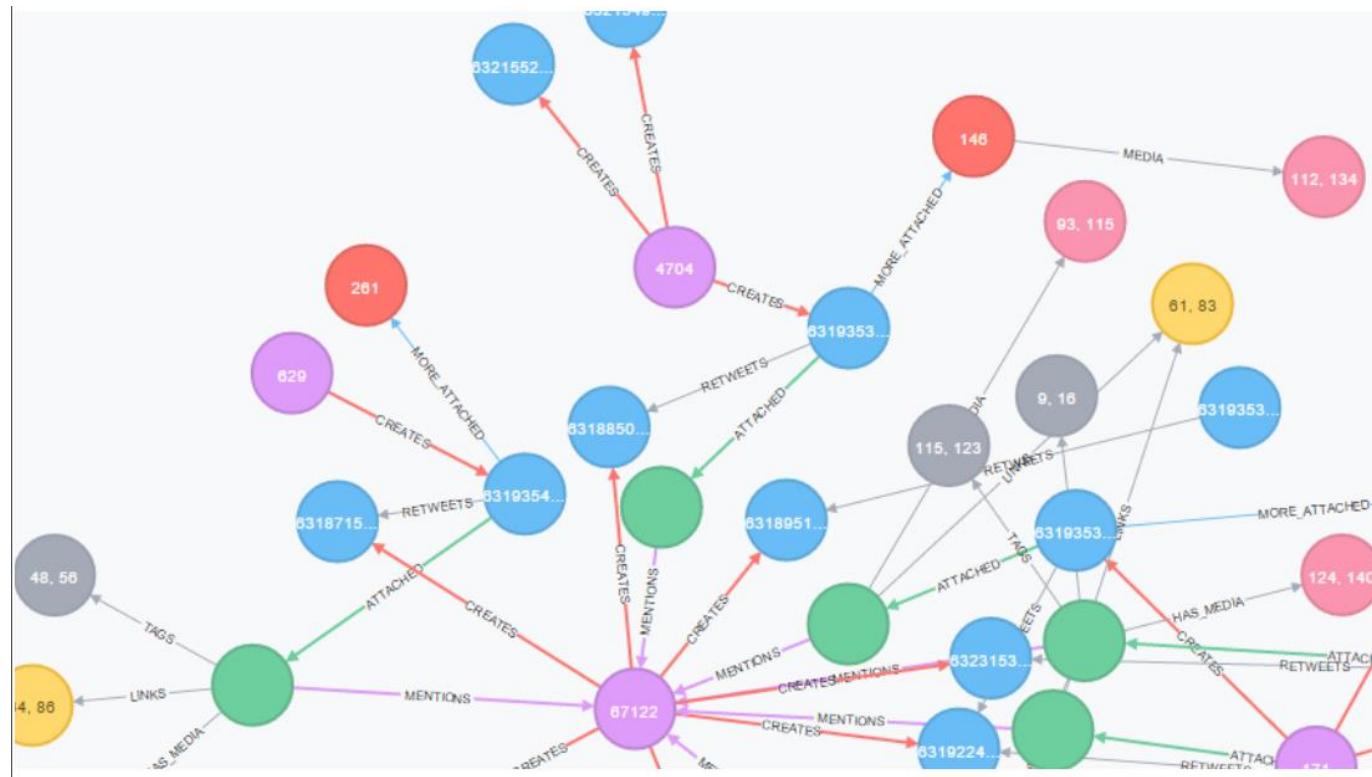


# Extensions: Multipartite networks:

## Use a graph database (e.g. Neo4J)

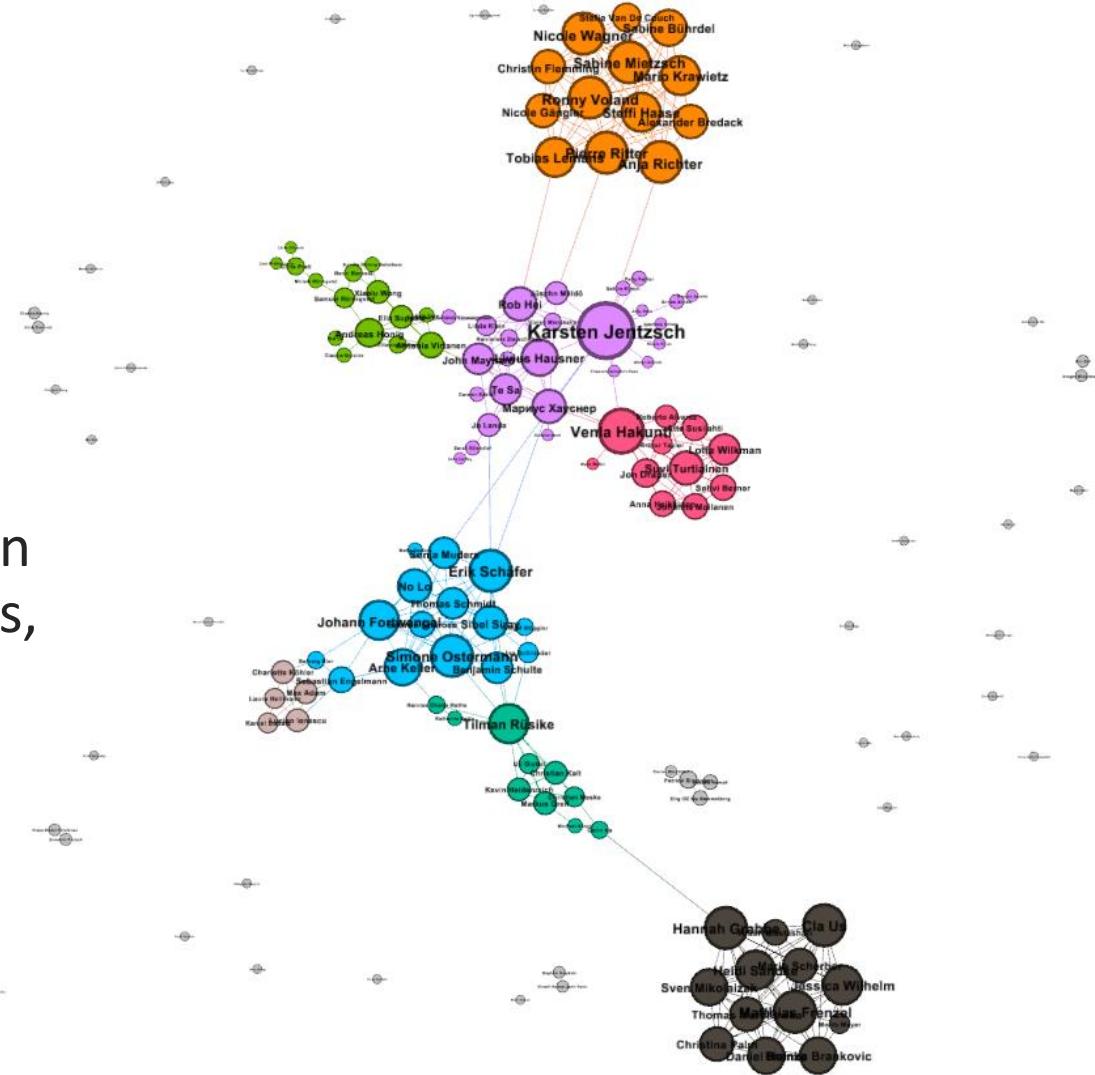
Many kinds of nodes: Users, Tweets, URLs, Hashtags, Media

Many kinds of edges: User creates Tweets, A Tweet is in response to another Tweet, A Tweet retweets another, User mentions User, Tweet contains Hashtag, User follows User



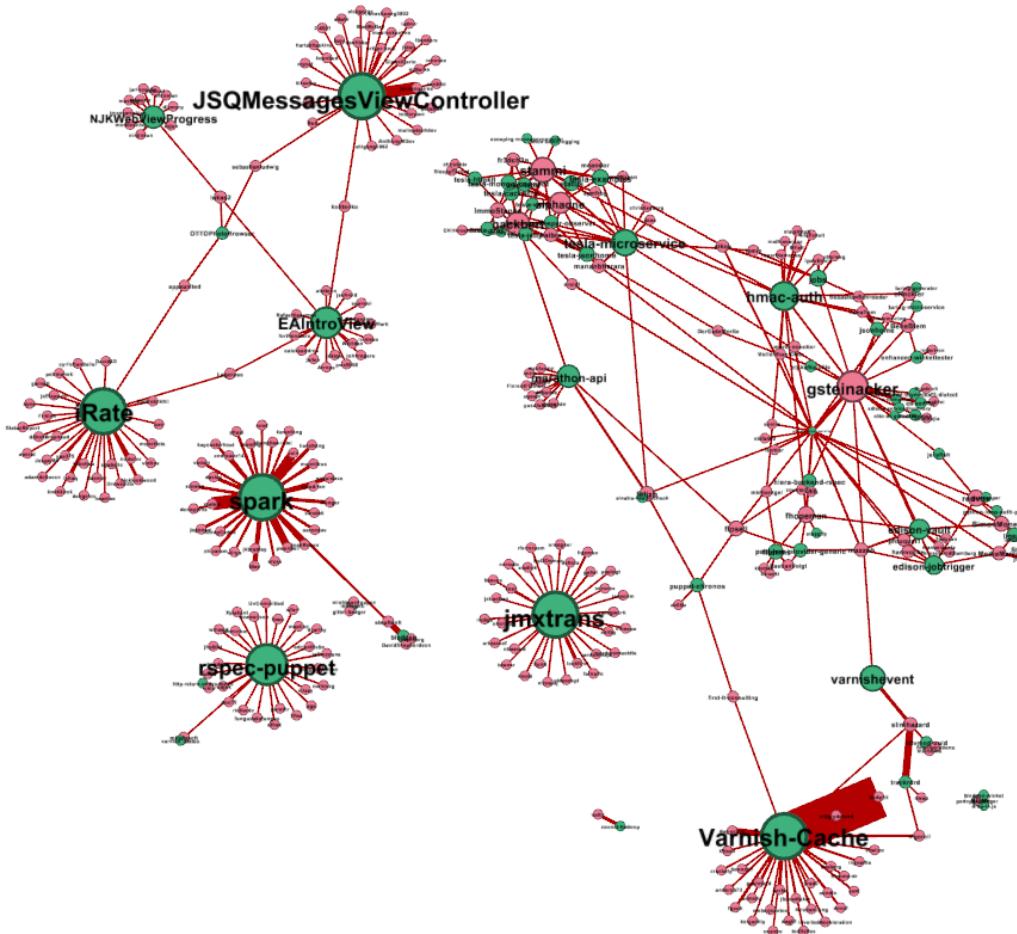
# One word about Facebook data...

- Facebook Graph API has been constantly closed currently very hard to extract relevant information
- Programs such as Netvizz help to extract information but limited to open groups, pages,...
- Possibly you have to use web scrapers to extract relevant data



# One word about Github data...

- GitHub is a good source for information about socio-technical systems
  - GitHub API reasonably easy to use to extract relevant information
  - Currently less restricted with regards to extracting data from previous points in time



## **Fig.1: Otto.de open source repositories & developer communities**

# Other potential sources

---

- [Programmable Web](#) provides information about available mashaups and API ecosystems
- [LinkedIn](#) has a python API and some tools to do automatic analysis (e.g. [Socilab](#))

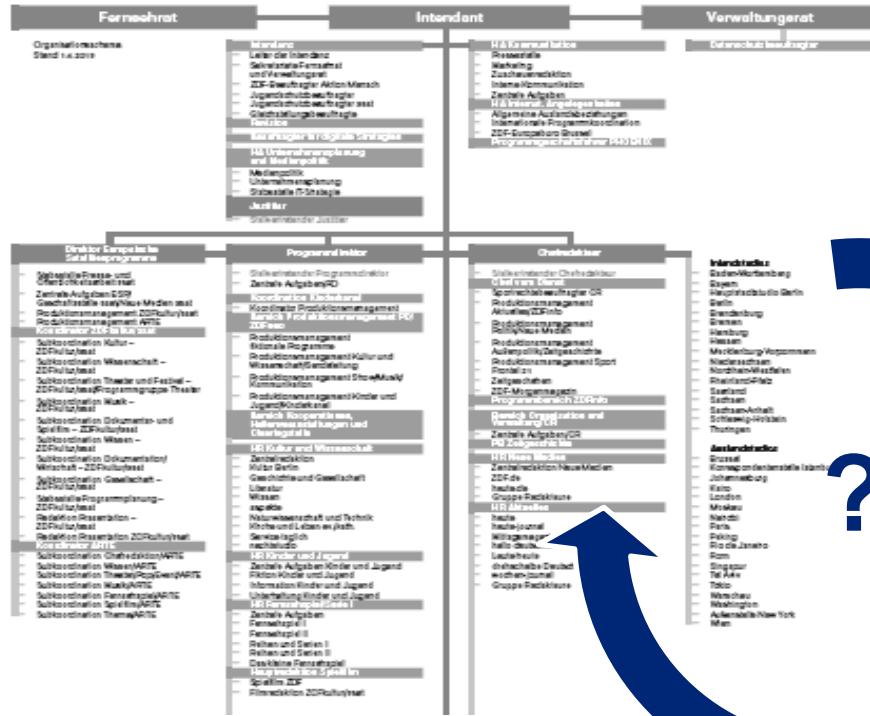
# Agenda

---

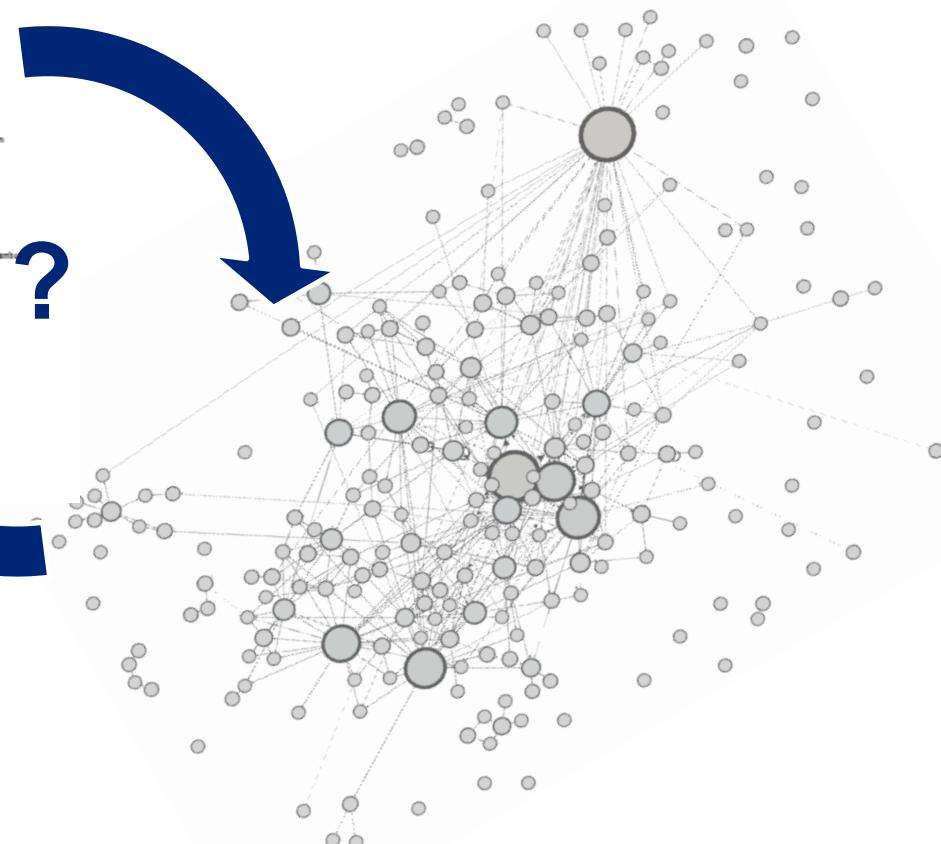
1. Introduction and examples
2. Fundamental concepts
  1. Creating graphs
  2. Visualizing graphs
  3. Centrality
  4. Community detection
3. Getting and handling data
4. Extended concepts
5. Discussion and conclusion

# Designing a network study: Motivation

## “Organizational Architecture”



## “IT Architecture”



# Dependent Variable: Application Governance (Horizontal Allocation of Decision Rights)

## 1. IT-owned applications (n=30)

- Planning decisions with central IT
- Change coordination responsibility within central IT

## 2. Business-owned applications (n=96)

- Planning decisions with a BU-IT
- Change coordination responsibility within central IT

## 3. End user-owned applications (n=73)

- Planning decisions with a business unit
- Change coordination responsibility with central IT

## 4. End user-managed applications (n=26)

- Planning decisions with a business unit
- Change coordination responsibility with a business unit



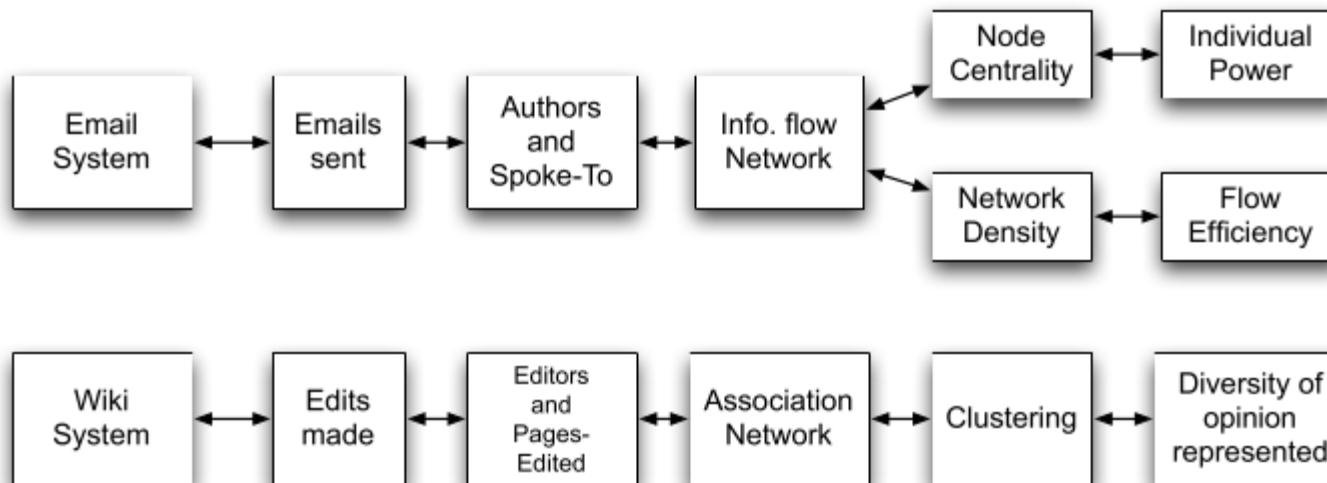
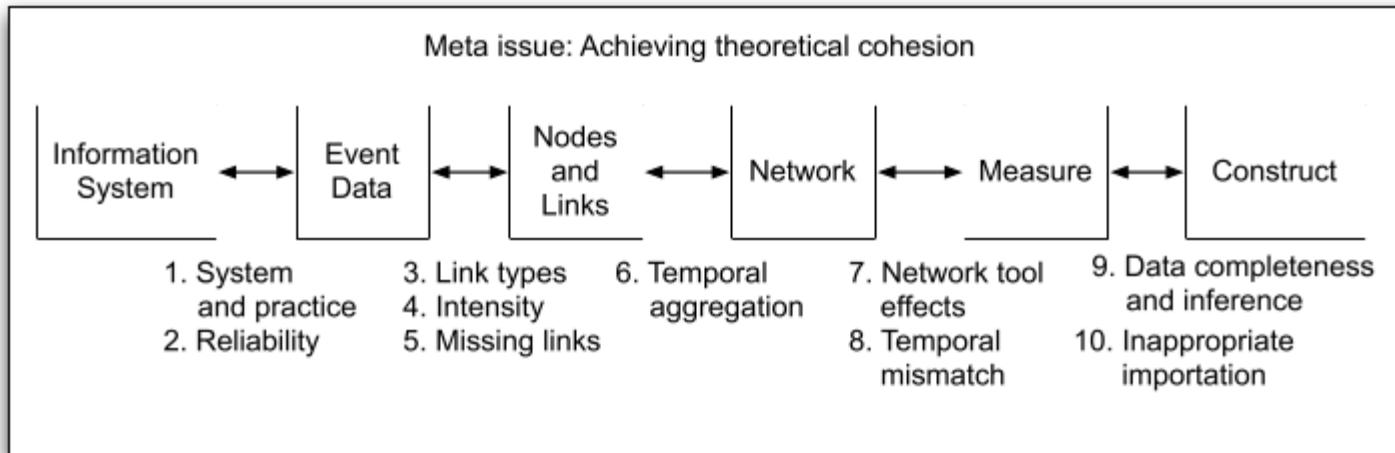
# Dependent Variable: Application Governance (Horizontal Allocation of Decision Rights)

	Est.	Sig.
Scope of Use	<ul style="list-style-type: none"><li>The breadth to which the system is used throughout the organization</li><li>Operationalization: #users (binned into 5 classes)</li><li>H1: Higher scope of use → more IT decision rights</li></ul>	<b>-.325</b> .002
Technical Specificity	<ul style="list-style-type: none"><li>The degree to which the system is customized to the organization</li><li>5 classes: cloud sw, standard sw, mod. ssw., custom-modular, custom</li><li>H2: Higher technical specificity → more IT decision rights</li></ul>	<b>-.189</b> .179
Positional Value	<ul style="list-style-type: none"><li>The number of other systems depending on this system</li><li>Operationalization: out-degree (numerical)</li><li>H3: Higher positional value → more IT decision rights</li></ul>	<b>-.098</b> .001
Neighbor Governance	<ul style="list-style-type: none"><li>The extent to which neighboring systems are planned by business or IT units</li><li>Avg. planning decision rights of direct neighbors in graph (numerical)</li><li>H4: decision rights of neighbors → decision rights (positive correlation)</li></ul>	<b>1.023</b> .000

# Conclusions & Next Steps

- Evidence for an application-level complementarity between „organizational architecture“ and “IT architecture”
- Linking two formerly separated strands of literature
  - IT Governance (traditionally on „monolithic“ org-level)
  - IT Architecture (young stream of quantitative network views)
- Practical and theoretical implications: More differentiated, pluralistic view at IT Governance
- Caveat: Lack of criterion / limited prescriptiveness
- Next steps
  - Qualitative analysis of within-case outliers
  - Exploring reasons for deviating from statistical patterns

# Chain of Reasoning and Validity Issues in Network Analysis with Digital Trace Data



[Howison et al. 2011]



# Discussion

---

# Agenda

---

1. Introduction and examples
2. Fundamental concepts
  1. Creating graphs
  2. Visualizing graphs
  3. Centrality
  4. Community detection
3. Getting and handling data
4. Extended concepts
5. **Discussion and conclusion**

# Exercise: Design your own network study.

---

# Discussion

---



# Department of Information Systems

## School of Business & Economics

### PhD Course on Social Network Analysis

Prof. Dr. Daniel Fürstenau  
20 February 2017