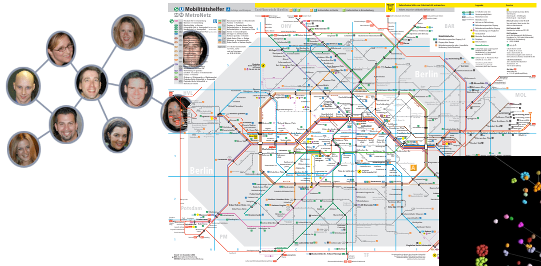
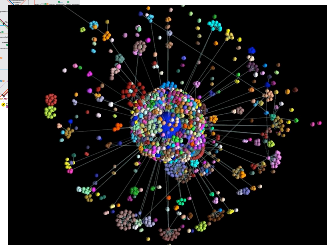
Characterization of Complex Networks



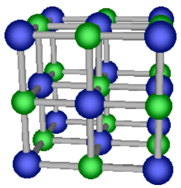


Gorka Zamora López
gorka(...)agnld.uni-potsdam.de
www.agnld.uni-potsdam.de/~gorka
Haus 22, Büro 0.32
0331 977-1821

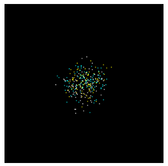
SUMMARY

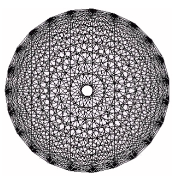
Complexity

Crystal Lattices

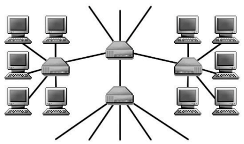


All-to-all interactions



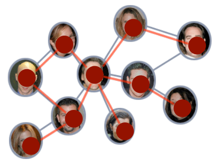


Internet

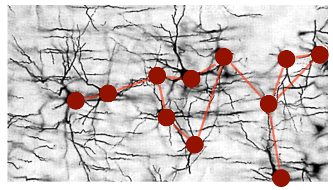


So... what is a network?

Social interactions

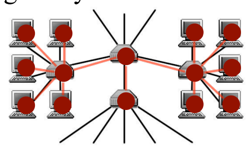


Biology



Transportation

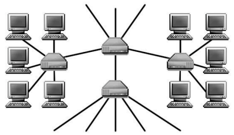
Technological systems



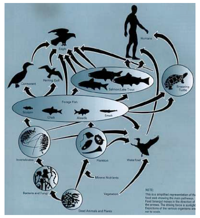
- Abstract manner to represent a large class of systems.
- Unique analysis toolkit: Graph Theory.

Network classes

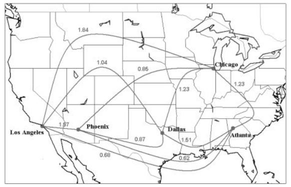
Graphs (undirected)



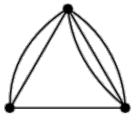
Digraphs (directed)



Weighted



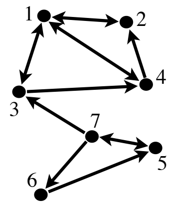
Multigraphs



Network representation

- *N*, number of nodes
- *L*, number of links

Density $\Rightarrow \rho = \dfrac{L}{N\,(N-1)}$ (no self-loops)



Adjacency Matrix

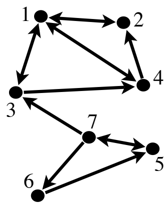
	1	2	3	4	5	6	7
1	0	1	1	1	0	0	0
2	1	0	0	0	0	0	0
3	1	0	0	1	0	0	0
4	1	1	0	0	0	0	0
5	0	0	0	0	0	0	1
6	0	0	0	0	1	0	0
7	0	0	1	0	1	1	0

Adjacency Lists

1 →	2	3	4
2 →	1		
3 →	1	4	
4 →	1	2	
5 →	7		
6 →	5		
7 →	3	5	6

Computational implications!

Degree distribution

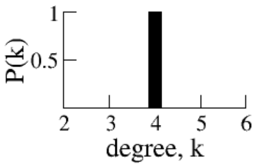


- Node degrees: $k = (k_i, k_o)$
  - node relevance
  - centrality
  - robustness

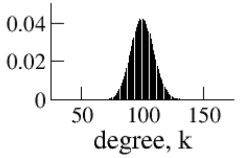
$$k_i = \sum_i A_{ij} \qquad k_o = \sum_j A_{ij}$$

- k – distribution

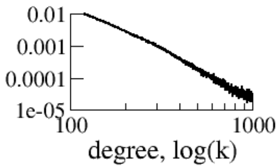
Regular Lattice



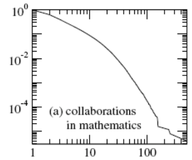
Random Network

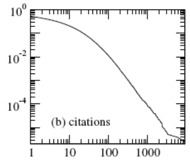


Scale-Free Network



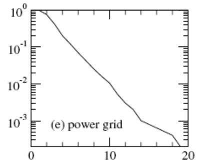
Scale-free networks

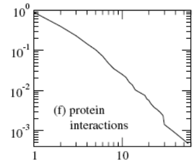


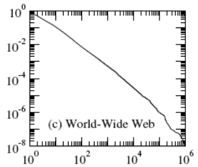


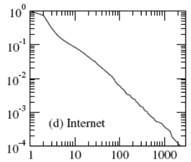
$$p(k) = k^{-\gamma}$$

- Robustness



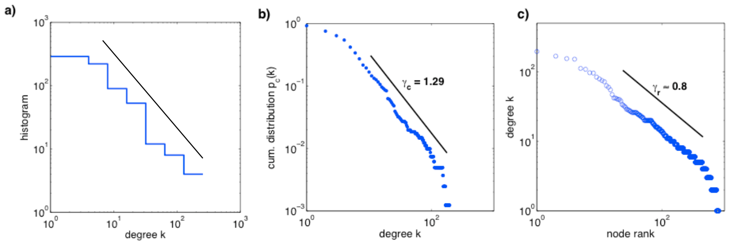






Scale-free networks

Binned data	Cumulative	Node rank

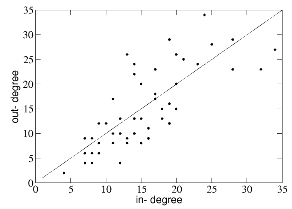


$$p(k) = k^{-\gamma}$$

$$p(k) = k^{-\gamma_c} : \quad \gamma = \gamma_c + 1$$

$$p(k) = k^{-\gamma_r} : \quad \gamma = 1 + \frac{1}{\gamma_r}$$

Degree correlations



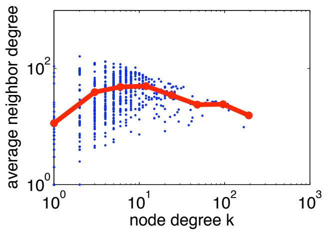
- 1-node correlations
- 2-node correlations: k ●——● q
  - Joint probability, *P(k,q)* Poor statistics
  - Pearson's correlation (for random links):

$$r = \frac{\sum_{k,q} kq(e_{kq} - a_k b_q)}{\sigma_a \sigma_b}, \quad -1 < r < 1$$

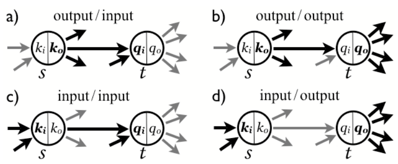
Only if linear relation!

Degree correlations

- 2–node correlations:
  - Neighbour's degree, $K_{nn}(k)$

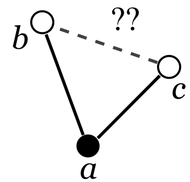


  - Directed networks



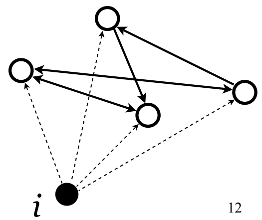
Clustering coefficient

Tendency for transitivity (cohesiveness) in a network: "are my friends also friends with each other?"



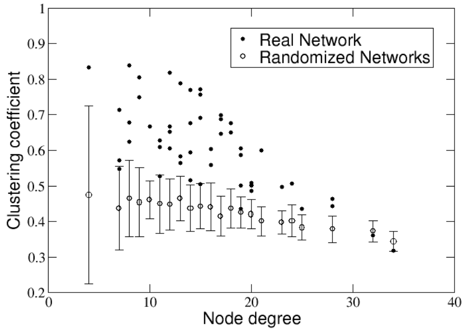
- Global clustering:
  - Strictly speaking… $P(bc|(ab \cap ac))$
  - Measuring… $C = 3 \times \frac{N(\bigtriangledown)}{N(\vee)}, \quad C \in [0,1]$
- Local clustering:

$$c_i = \frac{\sum_{j,k}^{N} A_{ij} A_{ik} (A_{jk} + A_{kj})}{k_i (k_i - 1)}$$



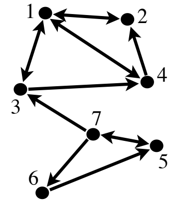
Clustering coefficient

- Average of $c_i \neq C$!!



Average c_i = 0.614
Clustering = 0.496

Distance



D_{ij} =

	1	2	3	4	5	6	7
1	2	1	1	1	∞	∞	∞
2	1	2	2	2	∞	∞	∞
3	1	2	2	1	∞	∞	∞
4	1	1	2	2	∞	∞	∞
5	3	4	2	3	2	2	1
6	4	5	3	4	1	3	2
7	2	3	1	2	1	1	2

- Path multiplicity → information transmission
- Average pathlength: $l = \langle D_{ij} \rangle$
- Graph diameter: $diam(G) = max(D_{ij})$
- Node centrality:

$$exc(i) = \frac{1}{N}\sum_j D_{ij} \qquad status(i) = \frac{1}{N}\sum_j D_{ji}$$

Data analysis, Overview

1) Observable X is measured → x
2) is x large or small? ⟶ interpretation
3) is x expected or surprising?
   3.1) significance testing
   3.2) re-interpretation.
4) other observables Y, Z …
   4.1) X, Y, Z correlations? ⟶ understanding!!

5) hopefully create a model !!

Significance testing

Compare the measured value, x, to the expected value of some random model.

- Analytical → *null-models with given conditions*

  Calculate expressions for:
  - expected value, $x_{expected}$
  - variance, σ

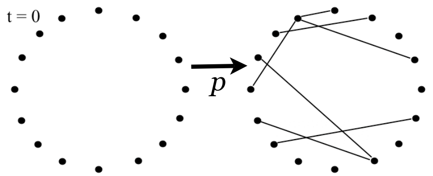
  } Usually very difficult !! :(

- Numerical → *ensembles of "random" networks*.

  For each realization measure X → x_i
  - ensemble average, $\langle x_i \rangle = x_{expected}$
  - deviations

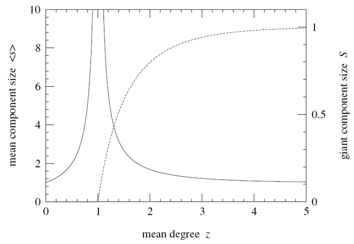
  } Computationally very demanding with large networks !!

Random Networks (Erdös - Rényi)

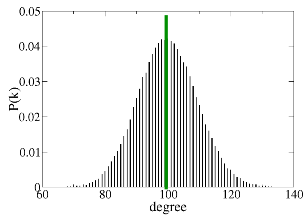


N nodes, L links

Percolation

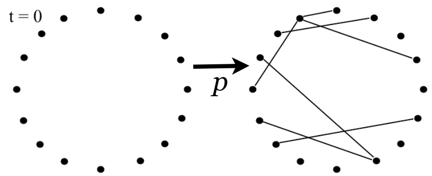


k-distribution



$z = \langle k \rangle = L/N$

Random Networks (Erdös - Rényi)



N nodes, L links

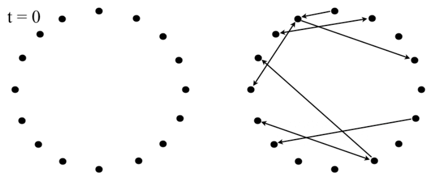
- Clustering coefficient

  C = link density = $2L / N(N-1)$

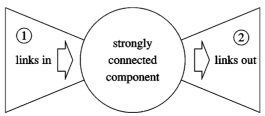
- Average pathlength

  $l \sim \ln N / \ln z$

Random Digraphs



- Percolation →

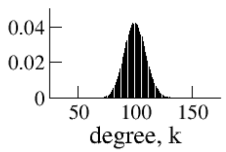


- Reciprocity → r = link density = $2L / N(N-1)$
- Clustering → No definition!!!

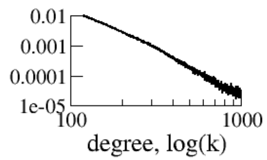
Arbitrary degree distribution

- What if a network has scale-free k-distribution?

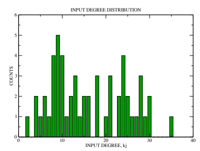
Random Network



Scale-Free Network



Real Network



- What if the data aquisition process introduces biases?

Arbitrary degree distribution

- Percolation threshold:
  - Undirected → $\sum_k k(k-2)p_k = 0$
  - Directed → $\sum_{k_i, k_o} (2k_i k_o - k_i - k_o) p_{k_i k_j} = 0$
- Reciprocity: → $r = \frac{L}{N^2} \frac{\langle k_i k_o \rangle^2}{\langle k \rangle^4}$

Arbitrary degree distribution

- Average distance
  - Undirected →
  - Directed →

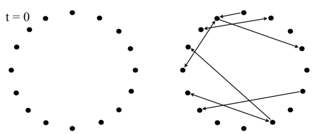
$$l = \frac{\ln(N/z_1)}{\ln(z_2/z_1)} + 1$$

- Clustering → ??!!

Even expected values of "simple" measures are very difficult to obtain analytically !!

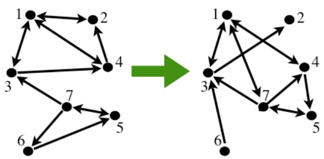
Ensembles of random networks

- 2 approaches:
  - Generation of networks from null sets



Usually faster!

  - Rewiring the real networks



Generation methods

- Random graphs/digraphs
  1) select two nodes at random: source s and target t
  2) link them: $s \rightarrow t$
  3) Watch out with self-loops and double links!

- With given k-sequences

  in-k: [1,1,1,1,2,3,3 … *N*,*N*,*N*]
  out-k: [1,2,2,2,3,3, … *N*,*N*]

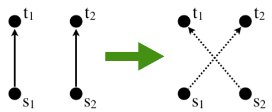
  → need tricks to avoid getting stucked

Rewiring methods

- Random graphs/digraphs
  1) select one link at random: $s \rightarrow t_1$
  2) rewire to another target: $s \rightarrow t_2$
  3) Watch out with self-loops and double links!

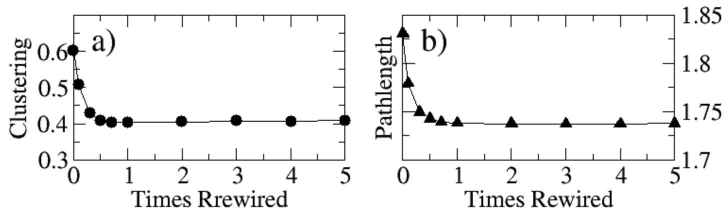
warning! how will you choose a link so that all links have same probability $1/L$ to be selected?

- Conserving the k-sequences
  1) select two links at random
  2) switch the links
  3) watch out with self-loops and double links!



Rewiring methods

- Need to test the validity of rewiring!!



Homework, before lab-day 1

- **Read** the introduction to Python. (www.python.org)
- **think**: having the adjacency matrix of a network, how will you calculate the degree of each node?
- **think**: how will you calculate the global clustering coefficient of a network? how do you calculate/count all paths of length = 2? how do you count the number of triangles?
- **think**: if you have the list with the degree of all nodes, how will you compute the cumulative degree distribution?

Homework, before lab-day 2

- **think**: how do you randomly introduce L links at random into an initially null matrix?
- **think**: how do you avoid the introduction of self-loops and multiple links?
- **think**: given a real network, how will you randomly choose a link? make sure that all links have the same probability of being chosen!!