CS-E5740

Complex Networks

Percolation, error & attack tolerance, epidemic models

Course outline

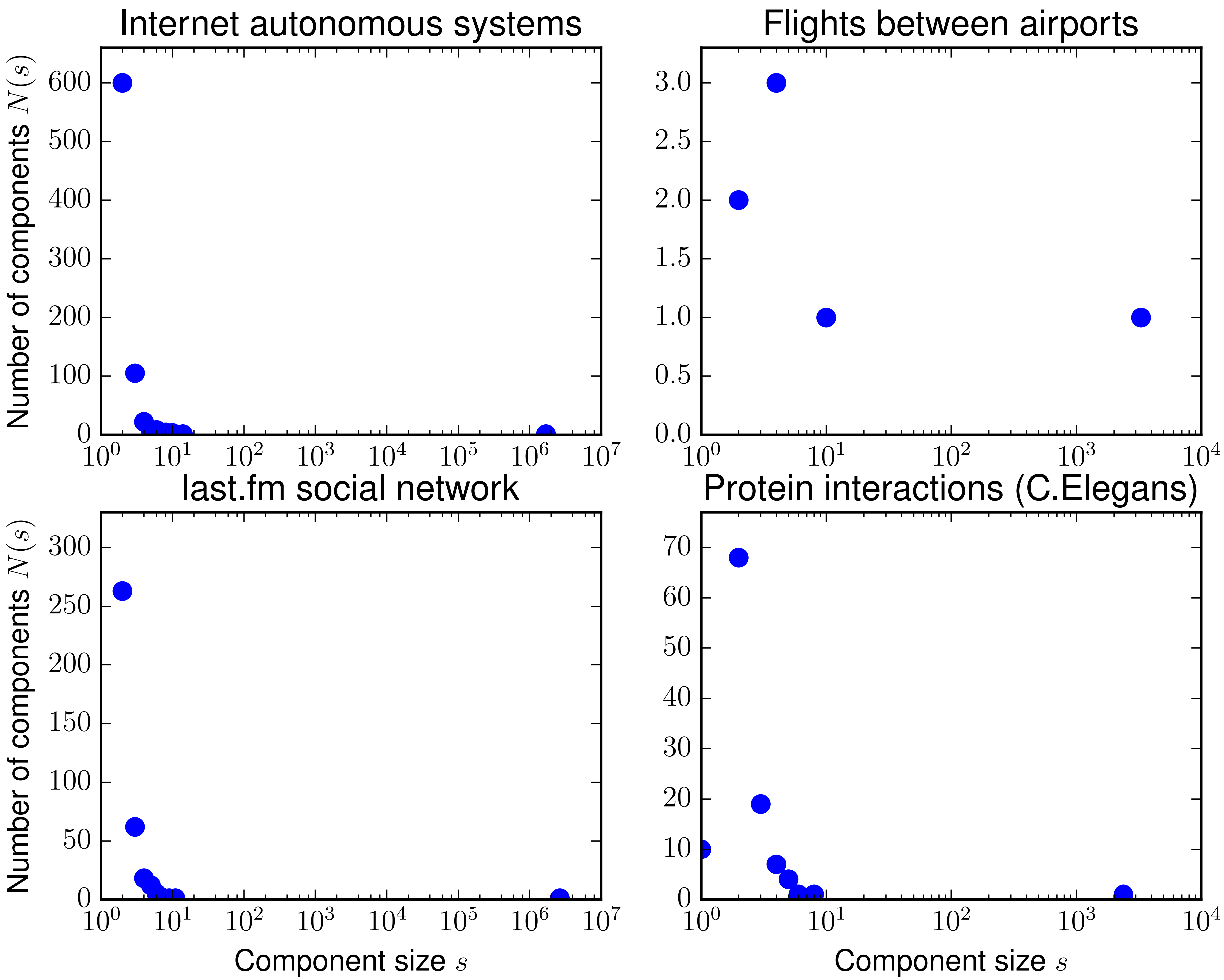
1. Introduction (motivation, definitions, etc.)
2. Static network models: random and small-world networks
3. Growing network models: scale-free networks
4. Percolation, error & attack tolerance of networks, epidemic models
5. Network analysis
6. Social networks & (socio)dynamic models
7. Weighted networks
8. Clustering, sampling, inference
9. Temporal networks & multilayer networks

From last week

Network	Degrees	Paths	Clustering
Real-world	Fat-tailed	short	high
Erdős-Renyi	Poissonian	short	low
Regular lattices	Fixed	long	high*
Configuration	Free to choose	short	low
Watts-Strogatz	Fixed to Poisson	short	high
Barabási-Albert	Power-law	short	low
Holme-Kim	Power-law	short	high

* (depending on layout & clustering measure)

Components in data



- Many small components
- One “giant” component

When is a network connected?

- How many edges/nodes can fail randomly before a communication/transport network breaks into small isolated parts?
- When are people in social networks connected by possible paths of disease or information spreading?
- What if edges/nodes are removed with some strategy instead of (uniformly) randomly?

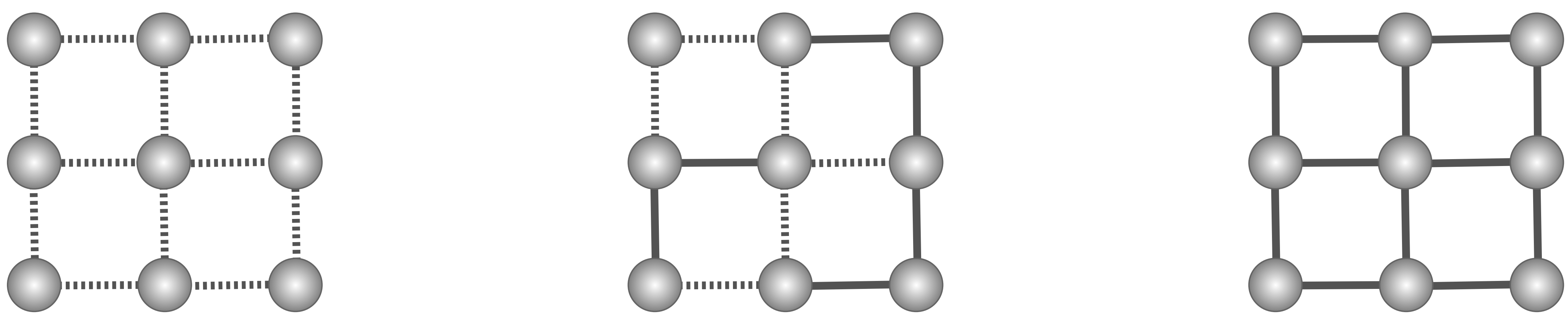
➡ Percolation theory

Percolation theory

- Theory of connectivity in very large systems developed in statistical physics
- The main observation: the system-level behaviour does not depend on small details in large systems but is “universal”
- Deep theory in physics and mathematics

Percolation theory

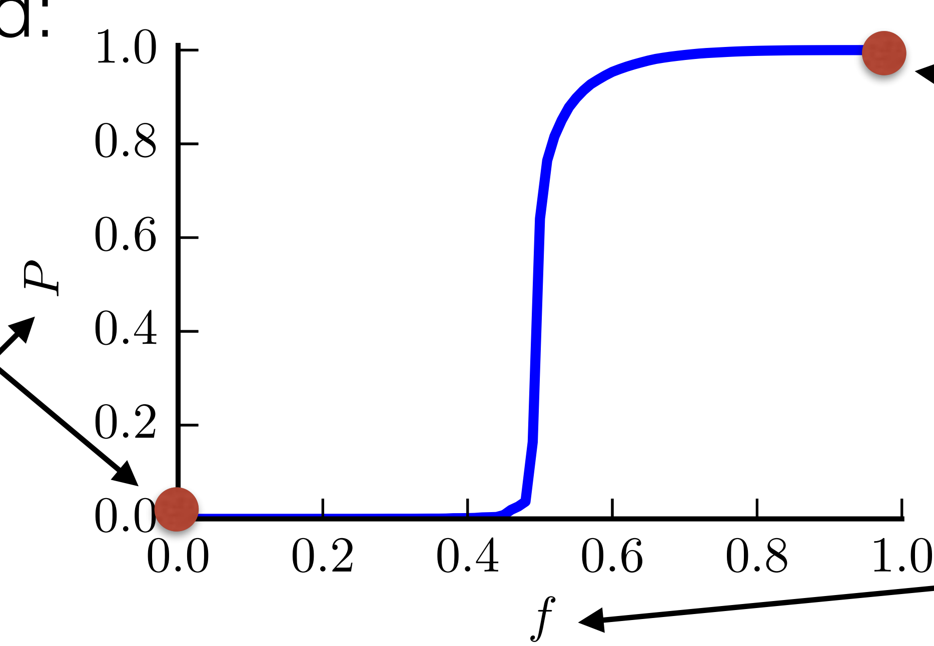
- Change something in the network (add/remove links, increase transmission probability, etc) and the component structure changes



Network disconnected:
f=0, P=~0

Network connected:
f=1, P=1

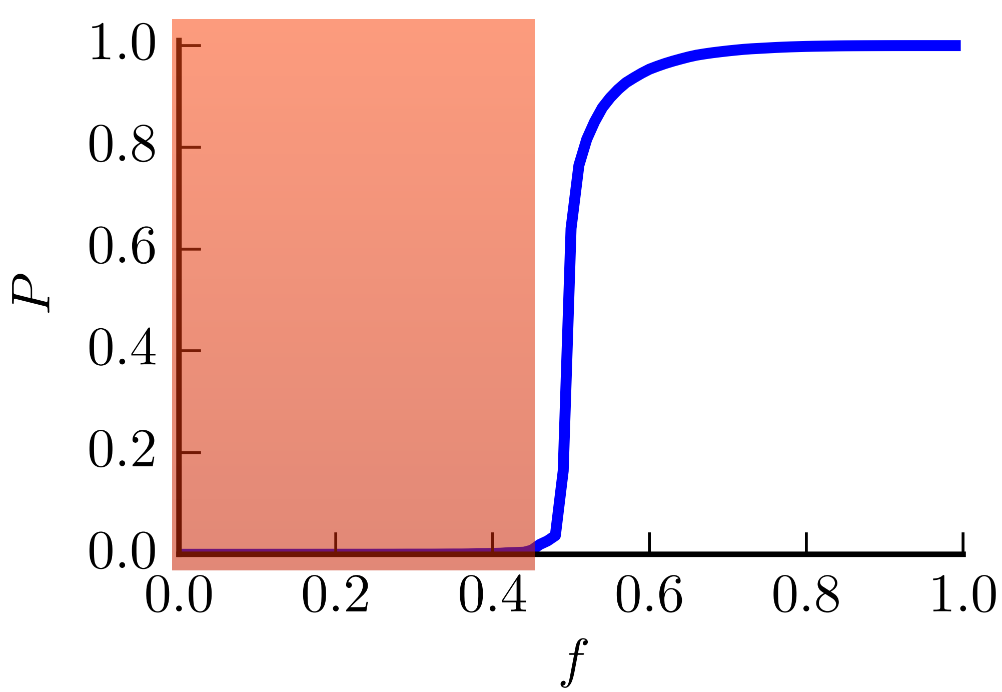
"order parameter" P:
fraction of nodes in
the largest connected
component (LCC)

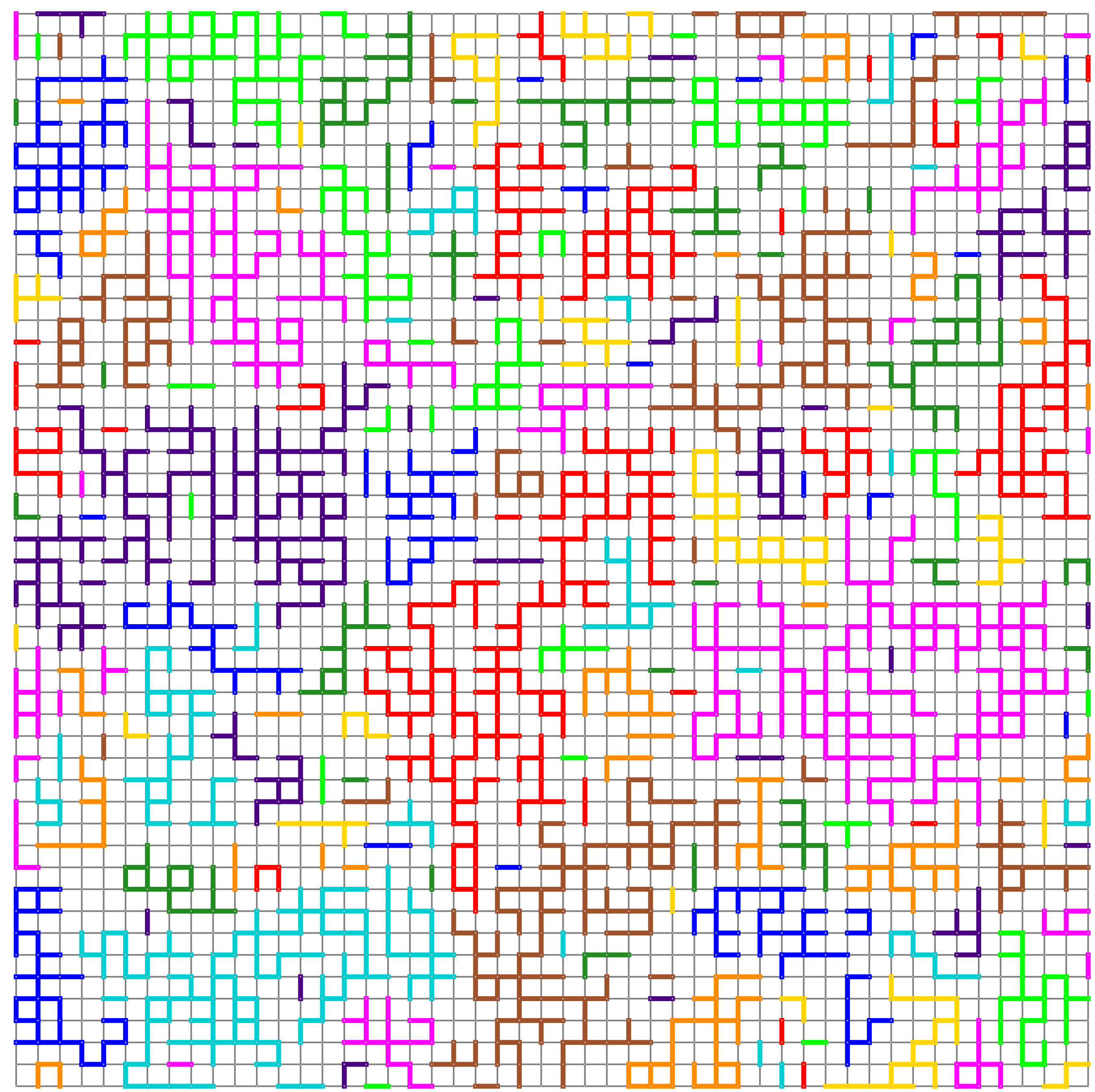


"control parameter" f:
fraction of active links

Disconnected phase

- Largest component relatively small
- Other components small

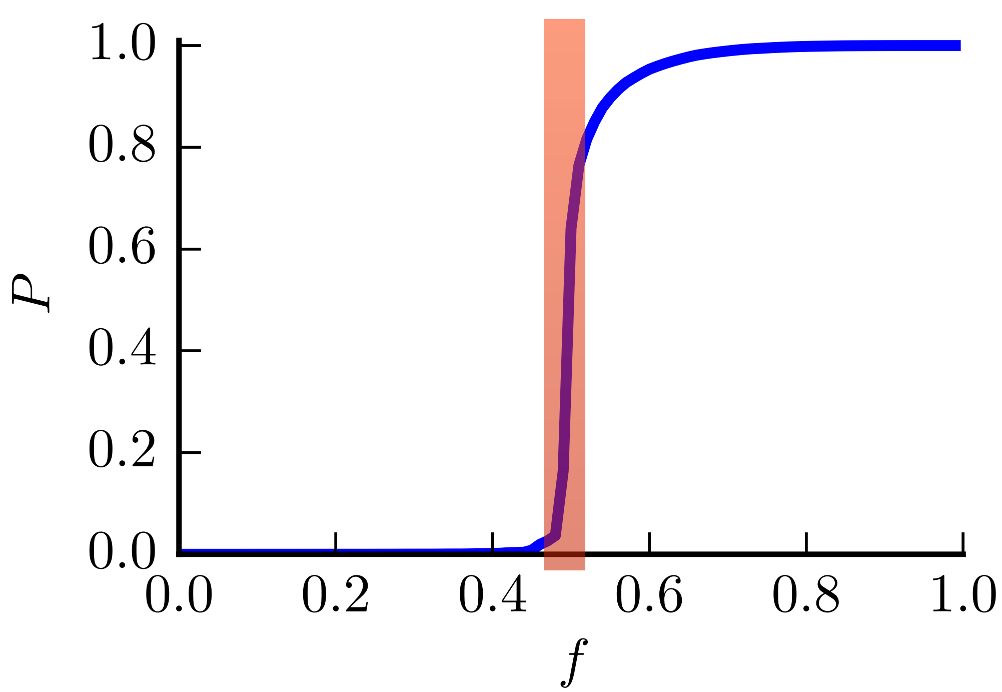


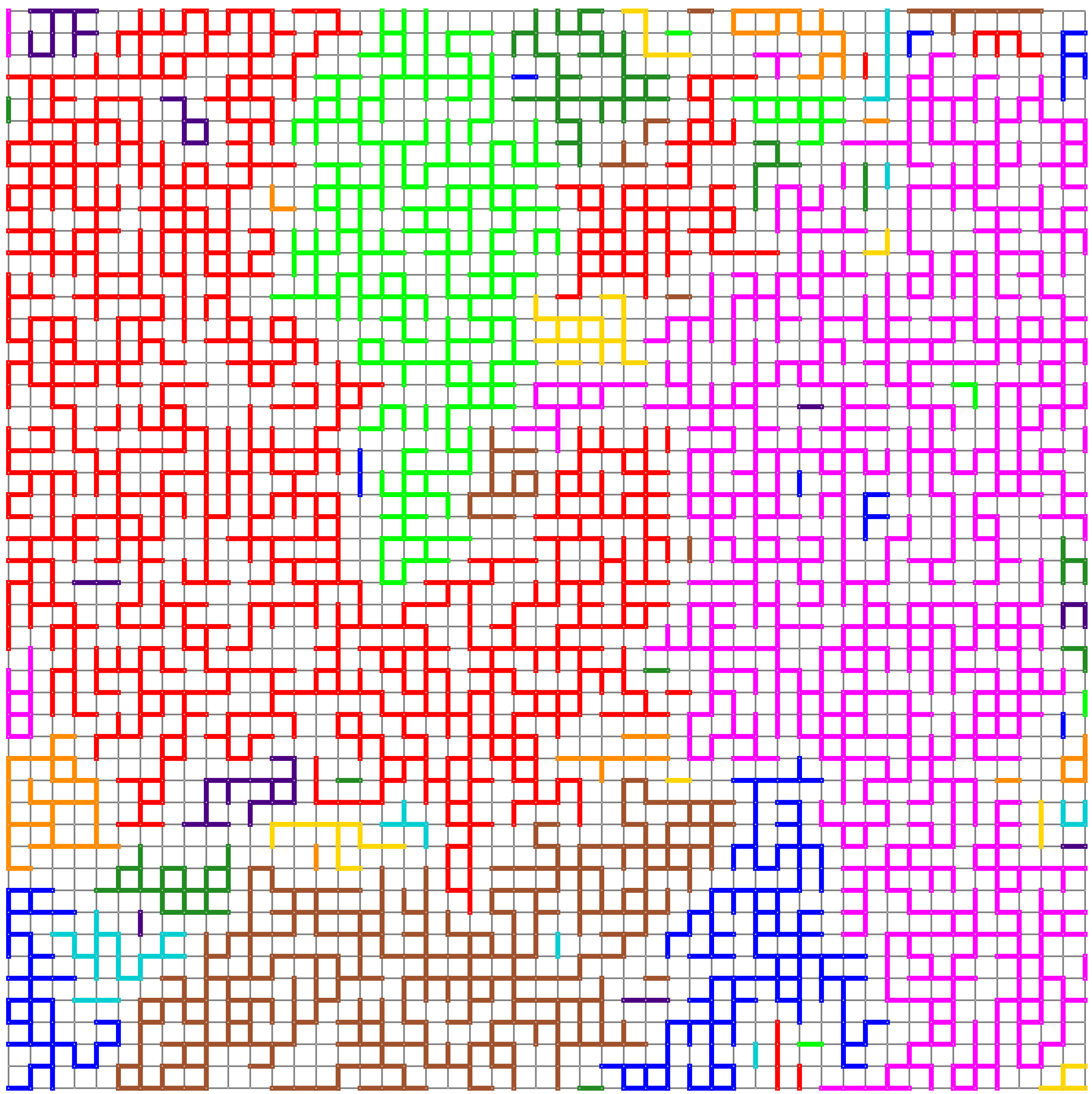


f=0.4

Phase transition

- The largest component becomes the "giant component"
- Other components from very large to small

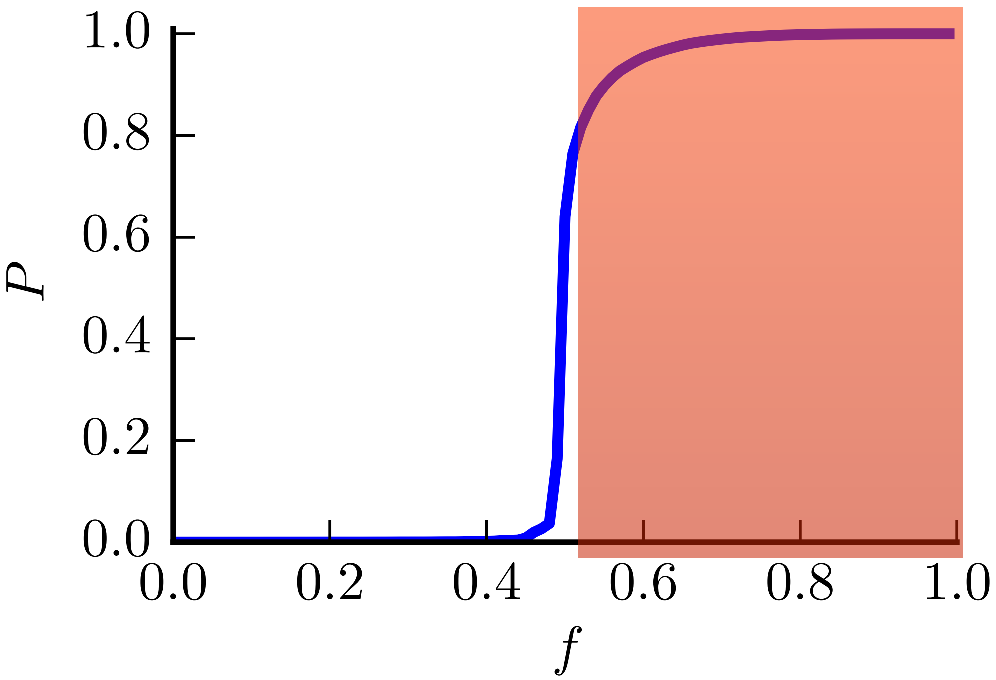


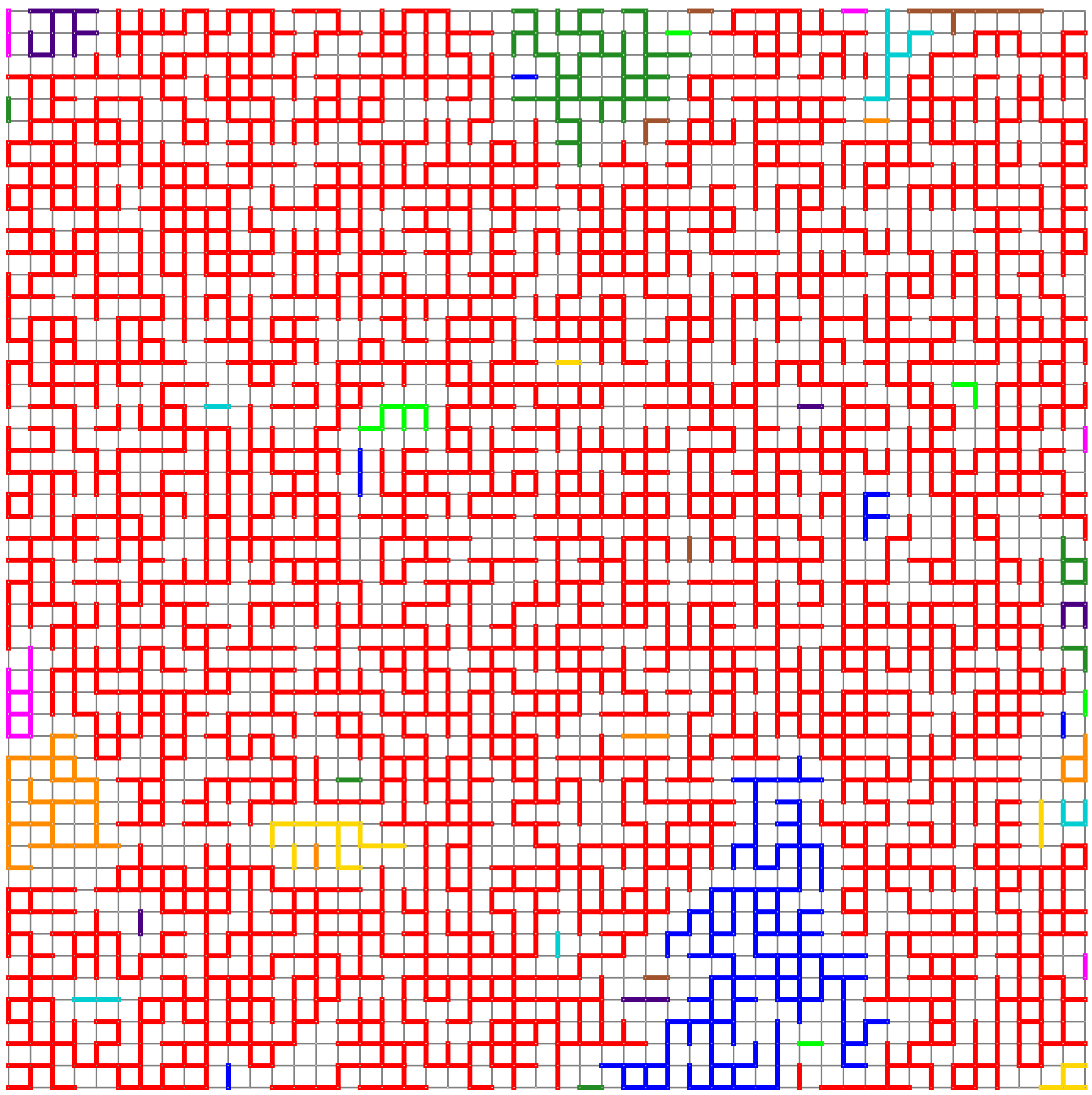


f=0.5

Connected phase

- The giant component size same scale as network size
- Other components small

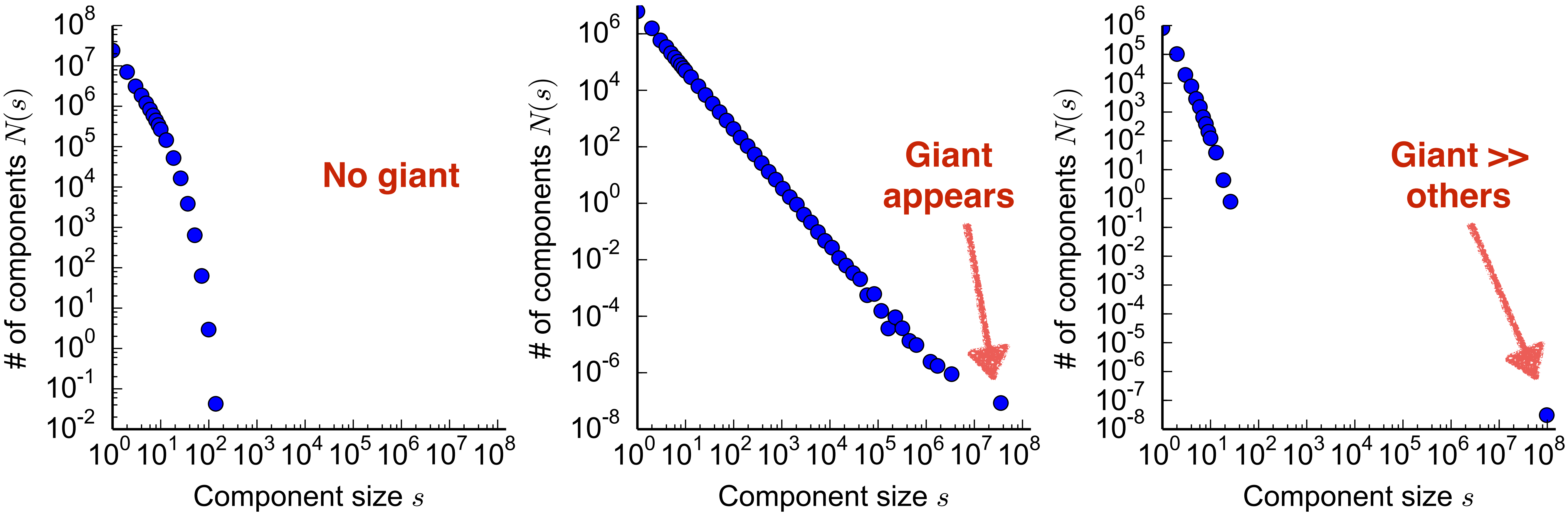




f=0.55

Component size distributions

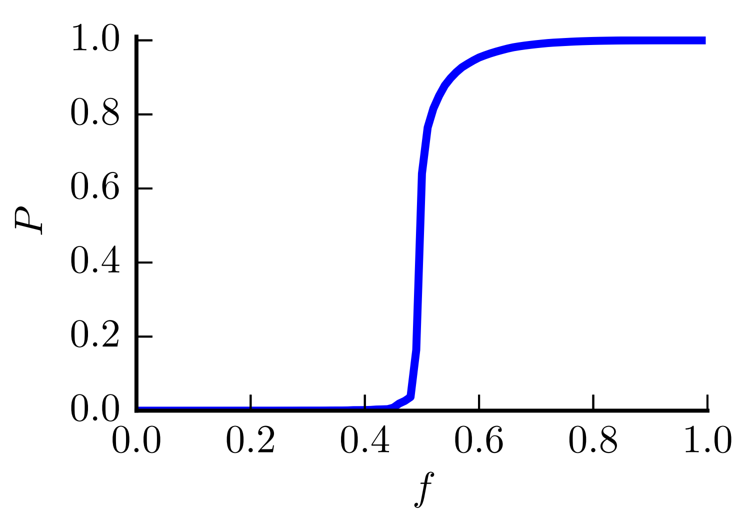
Disconnected | Phase transition | Connected

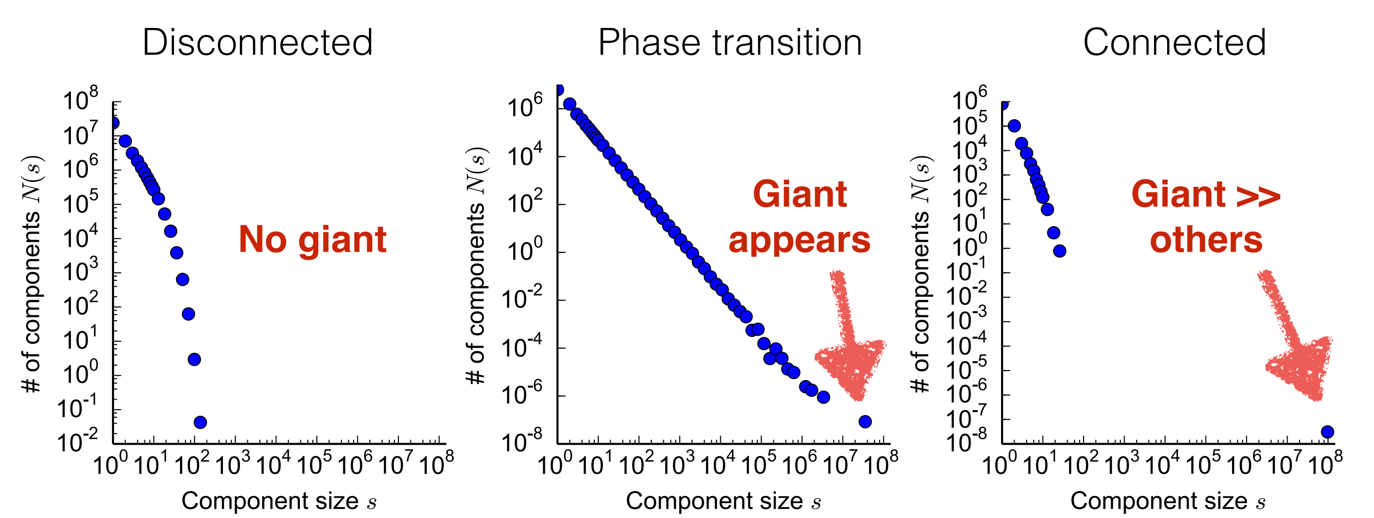


- The size distribution of other components in phase transition point a power law
  - “Critical point” in the theory of critical phenomena

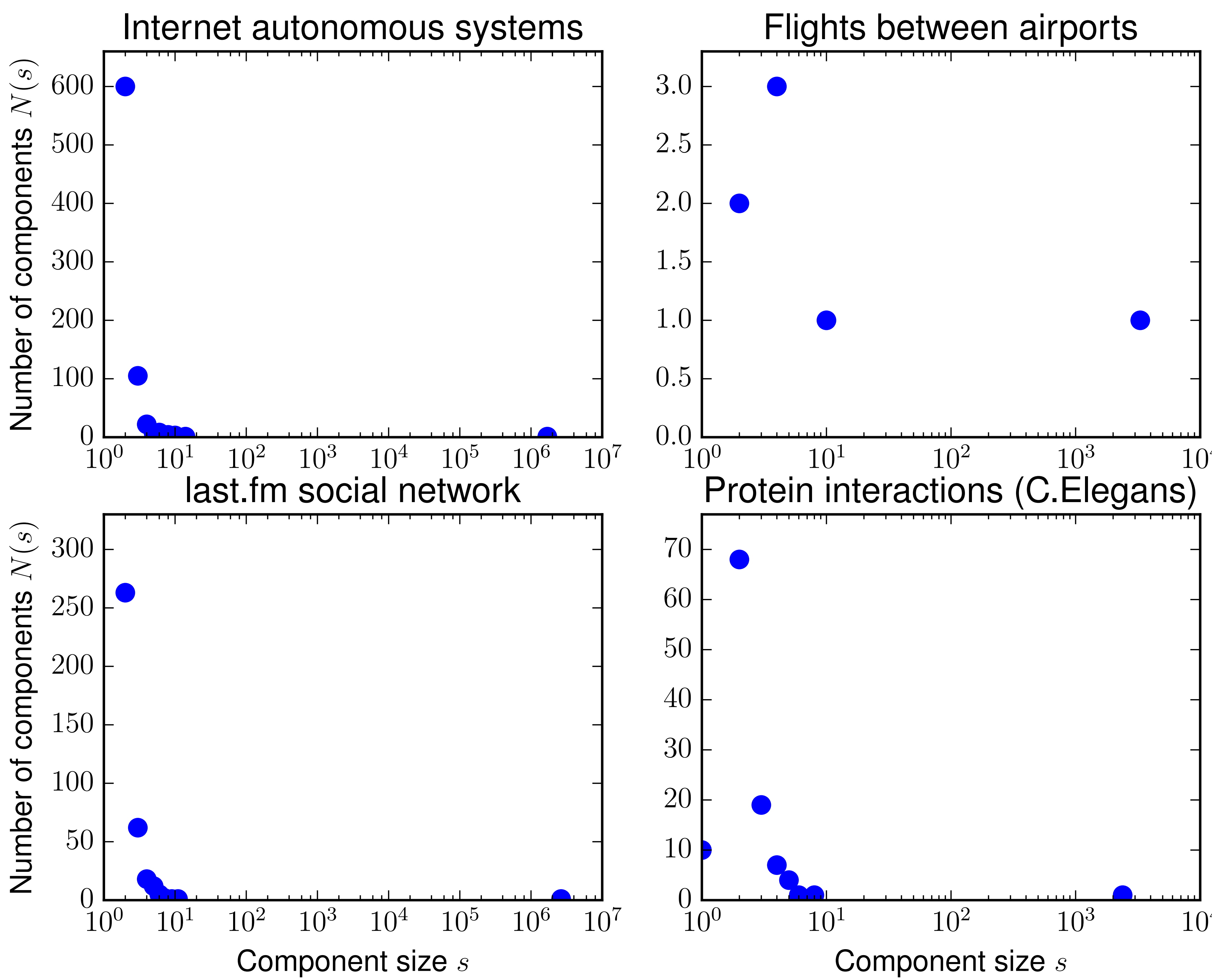
What is going on?

- “Rich get richer” phenomenon: large components are more likely to be connected to other components and grow in size
  - Disconnected: All components similarly small
  - Transition: Power-law size distr. = signature of rich get richer systems
  - Connected: The giant component is much bigger than all other components and likely to engulf all others



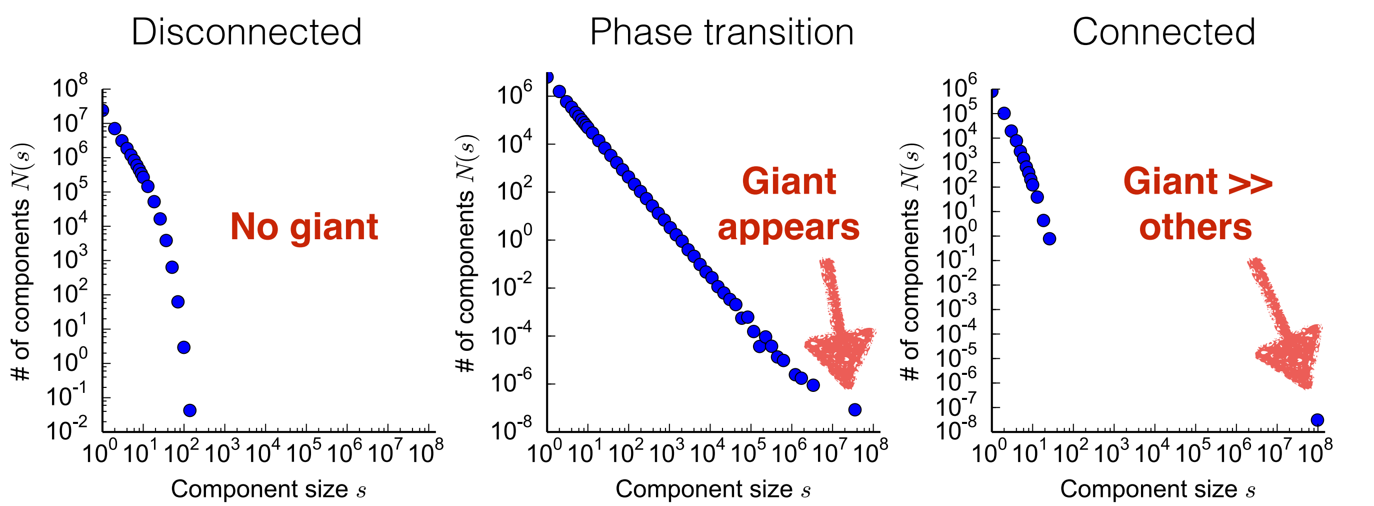


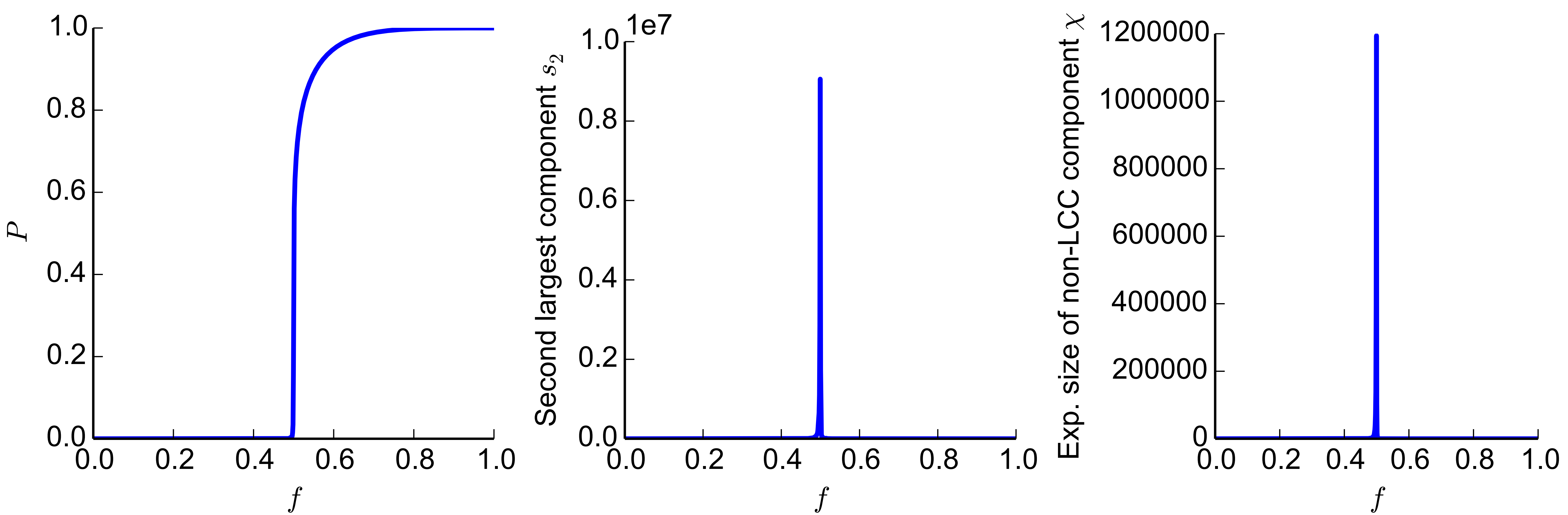
Components in data



Component size distributions

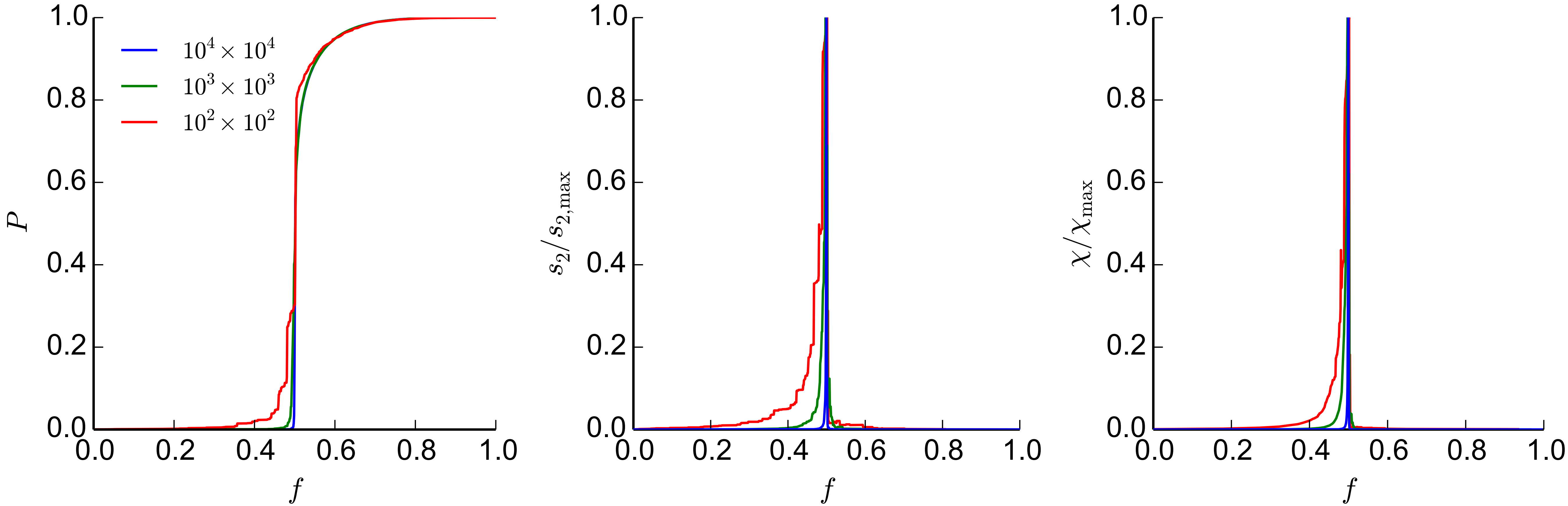
- The component size distribution of all other components except the largest can be used to find the phase transition point (i.e., critical point)



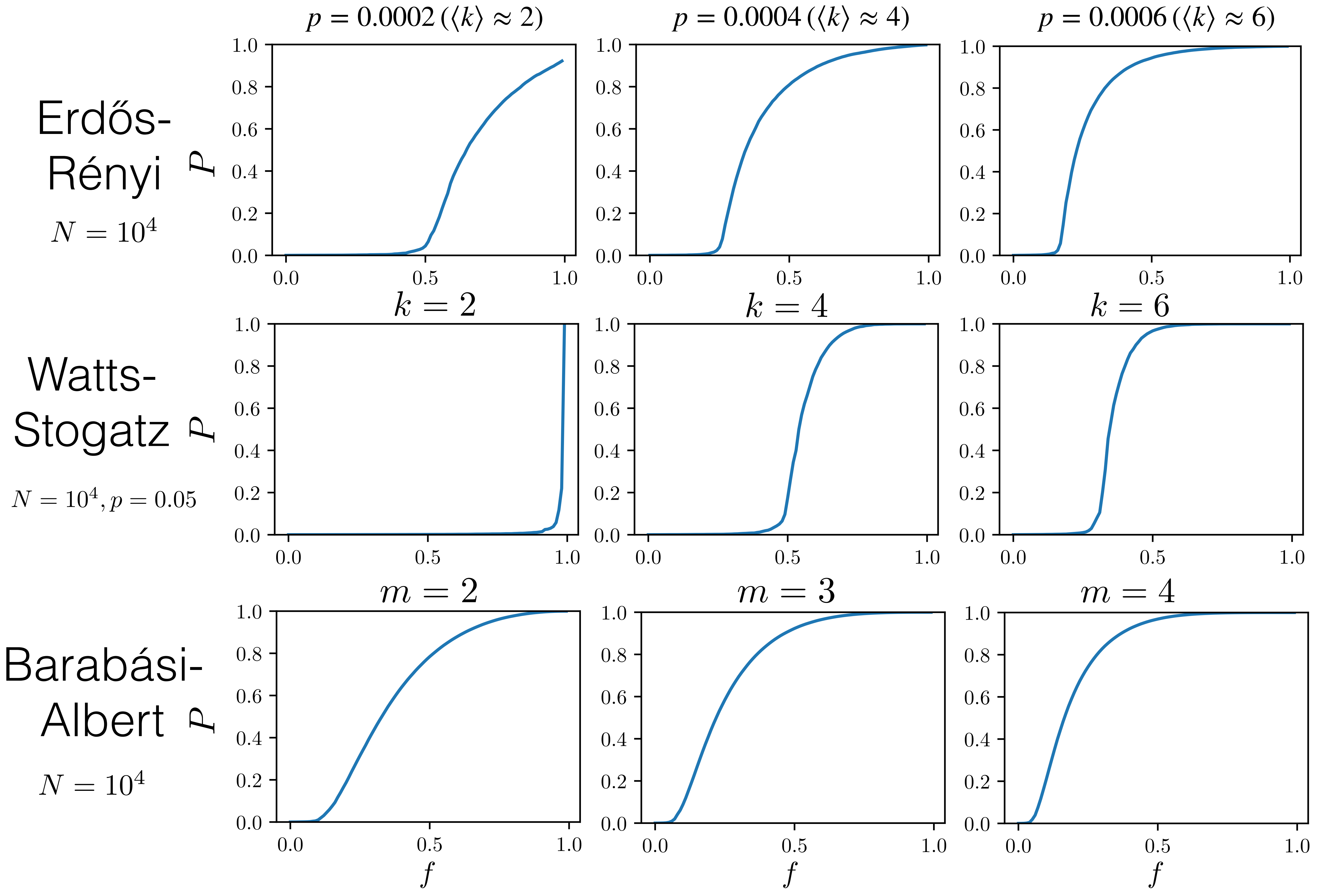


System size matters

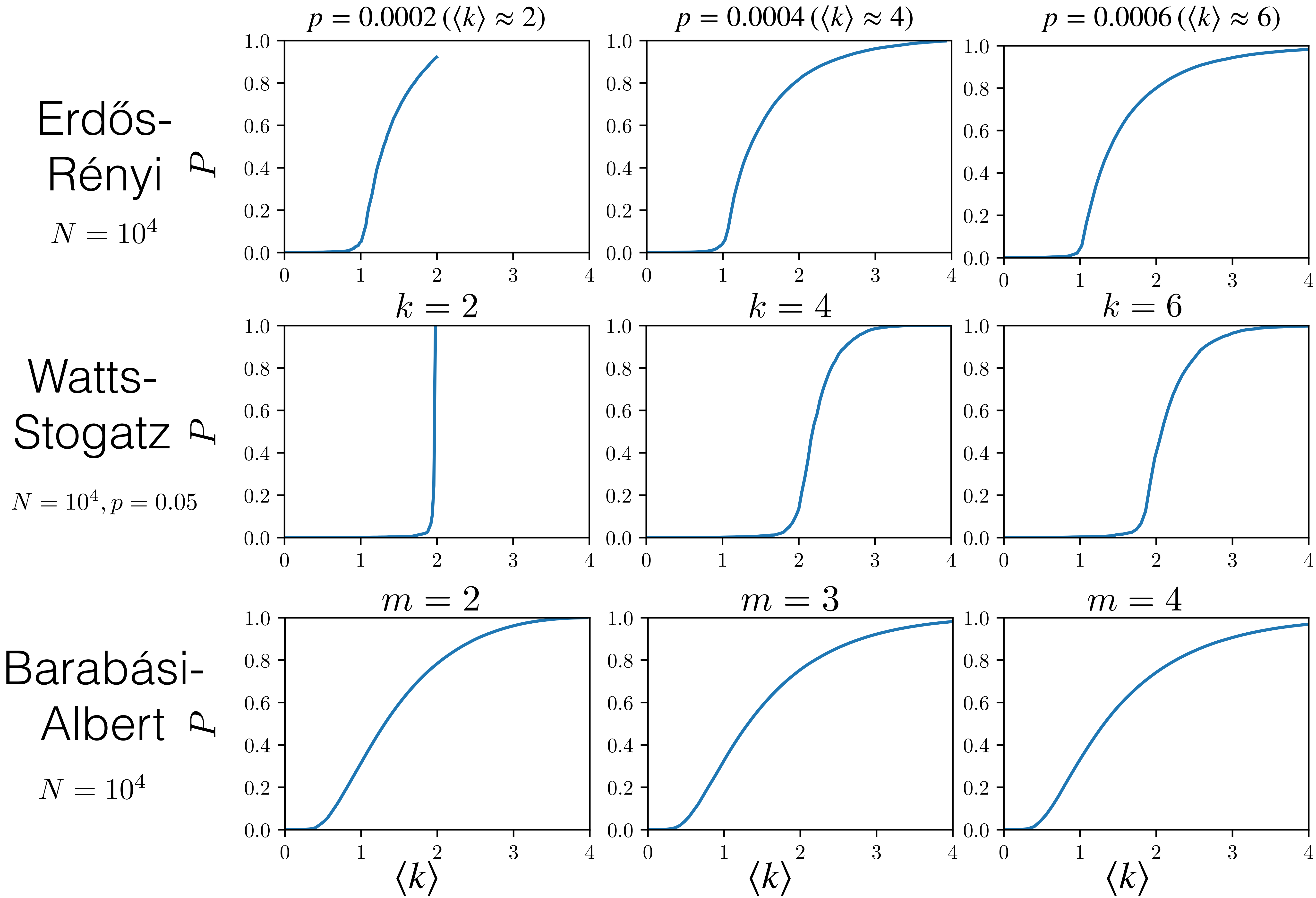
- Larger systems show clearer results
  - Faster transition, sharper peaks of 2nd component, etc.
  - Deviations from the infinite system size know as “finite size effects”



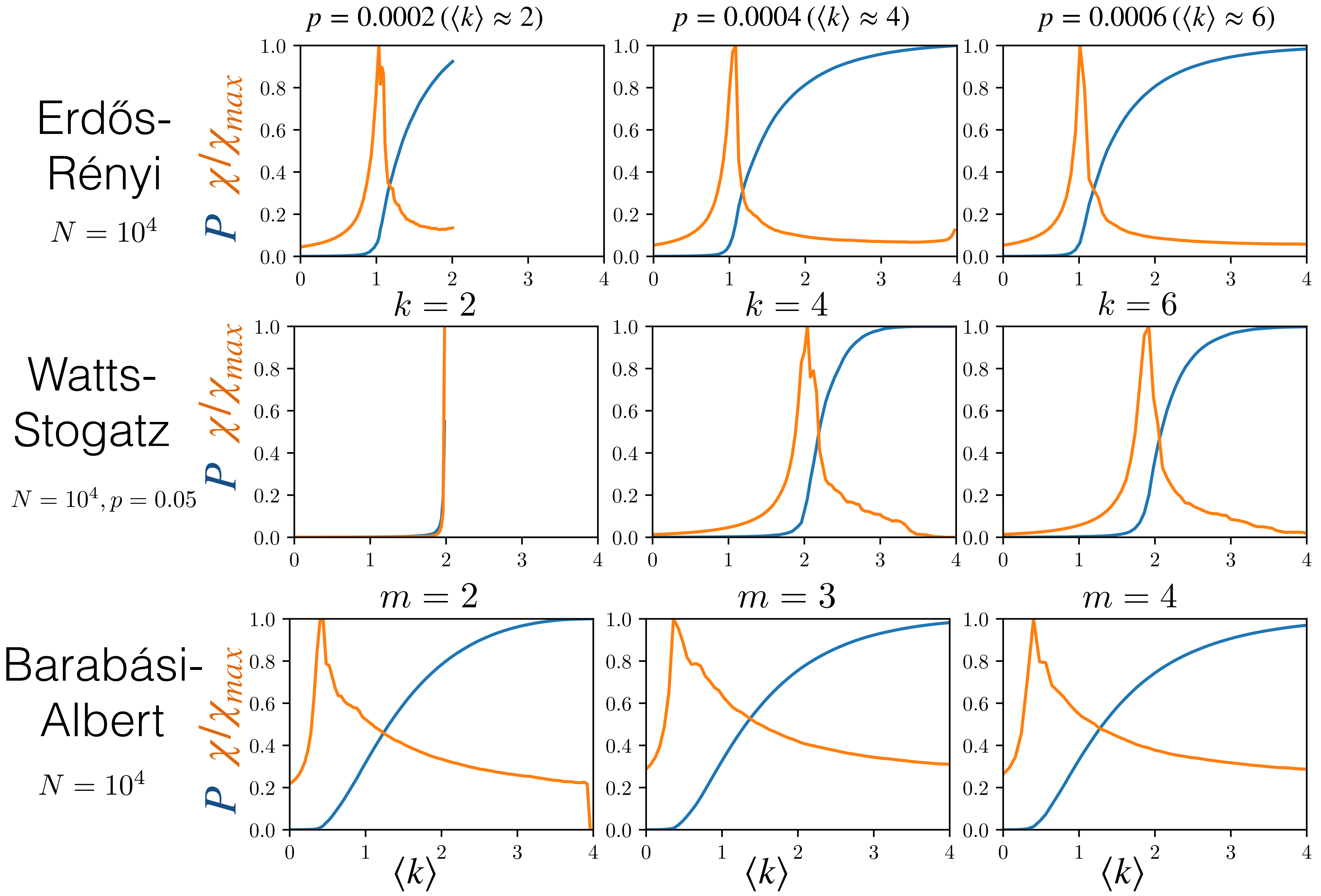
Some results for other models



Rescaling the x-axis to use avg degree $\langle k \rangle = 2fL/N$



Rescaling the x-axis to use avg degree $\langle k \rangle = 2fL/N$

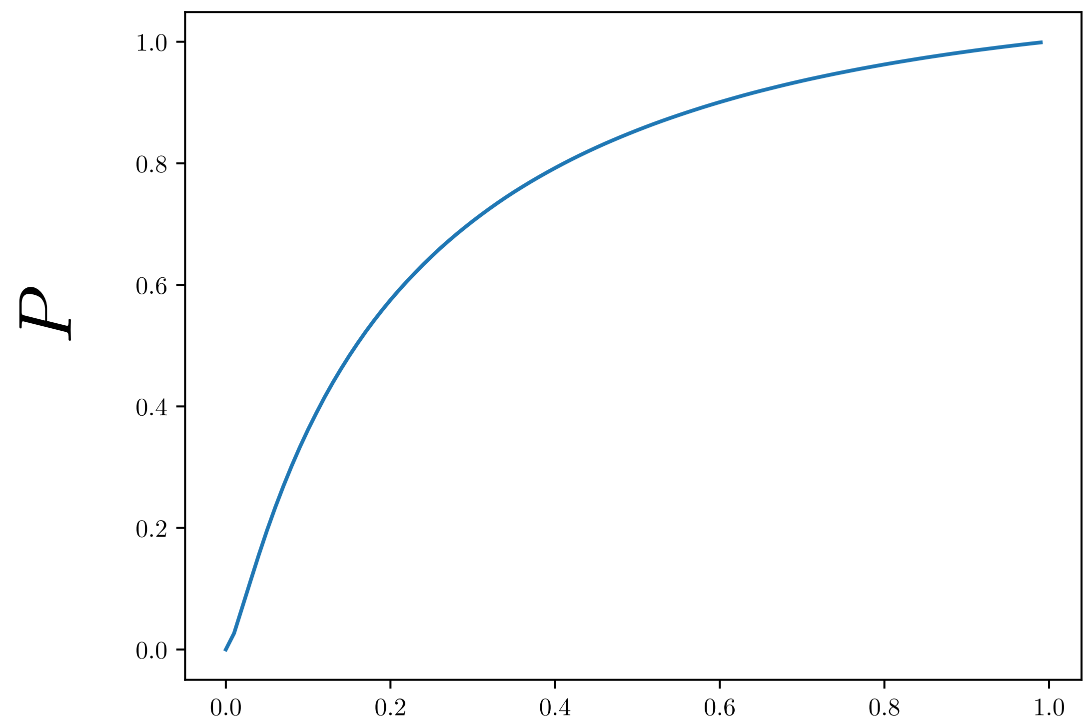


Erdős-Rényi networks & percolation

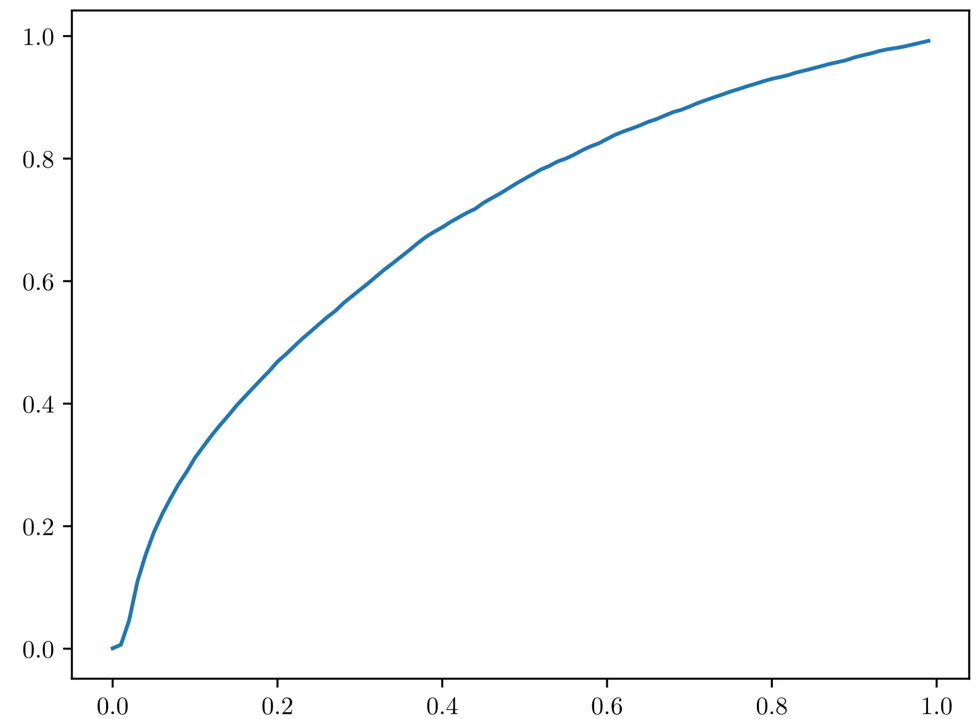
- Start from a full network
- Fraction of active links f (or remove 1-f fraction of links)
  - ➡ Roughly the same as ER network with p = f
- Remember <k> = (N-1)p, so instead of f we can use <k>=(N-1)f in the x-axis of the plots
- Similarly, for configuration model the degree distribution is modified when edges are removed, but result is another configuration model

Results for data

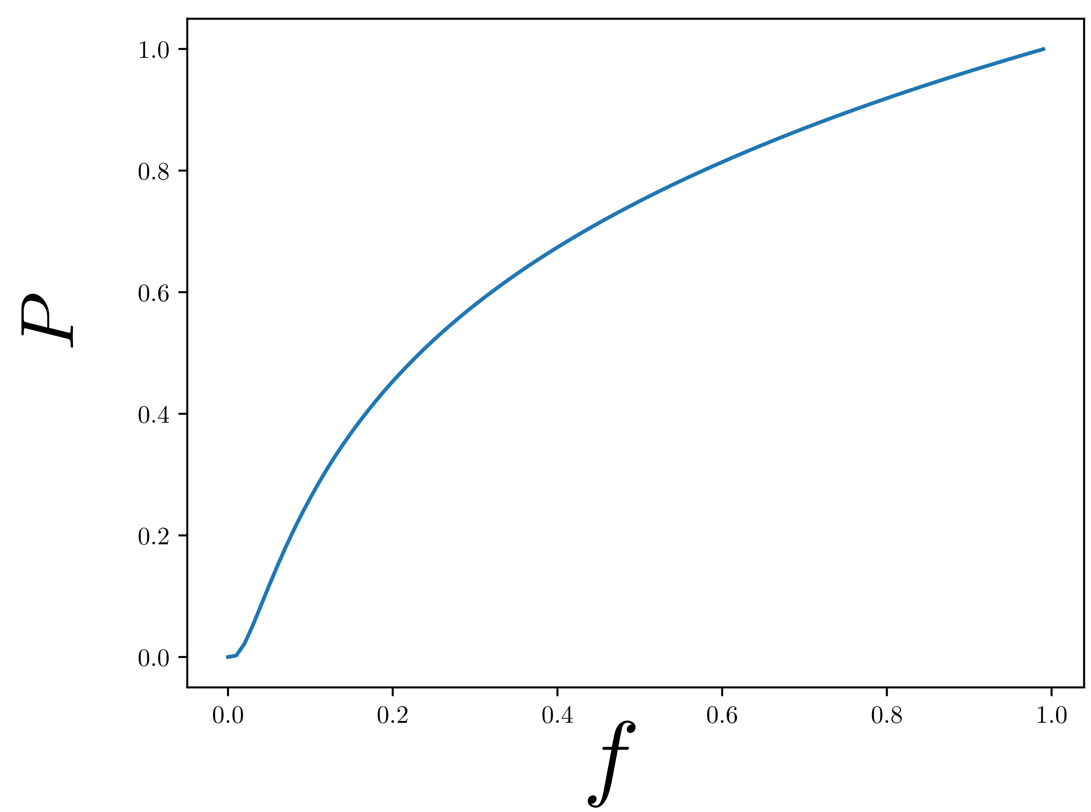
Internet autonomous systems



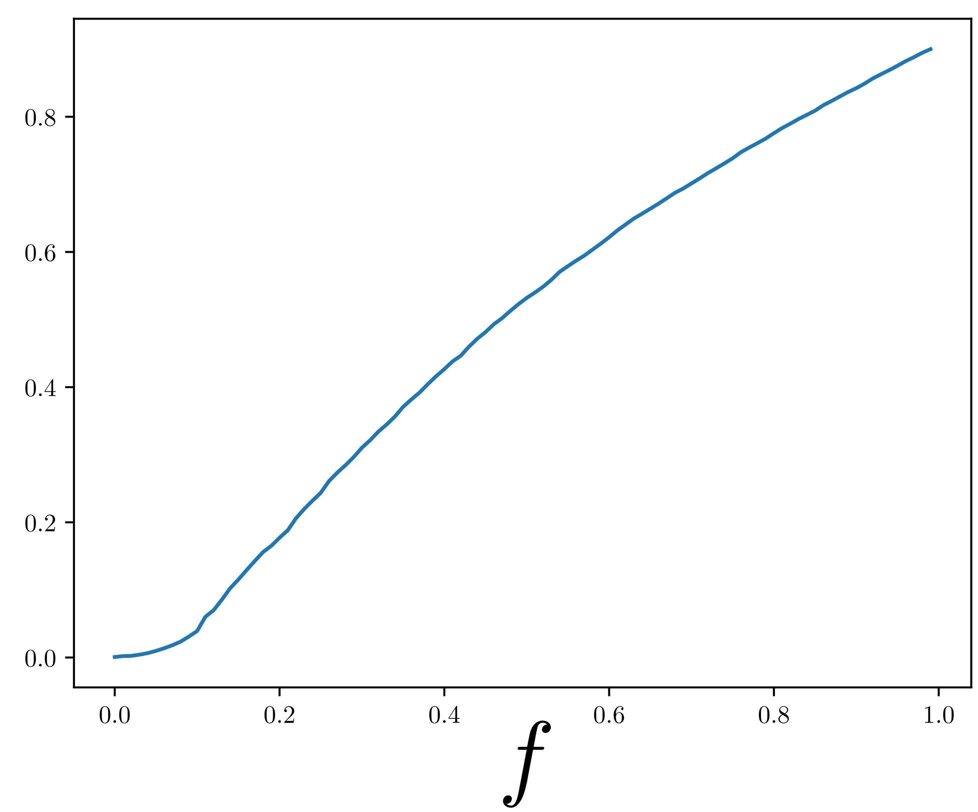
Flights between airports



last.fm social network

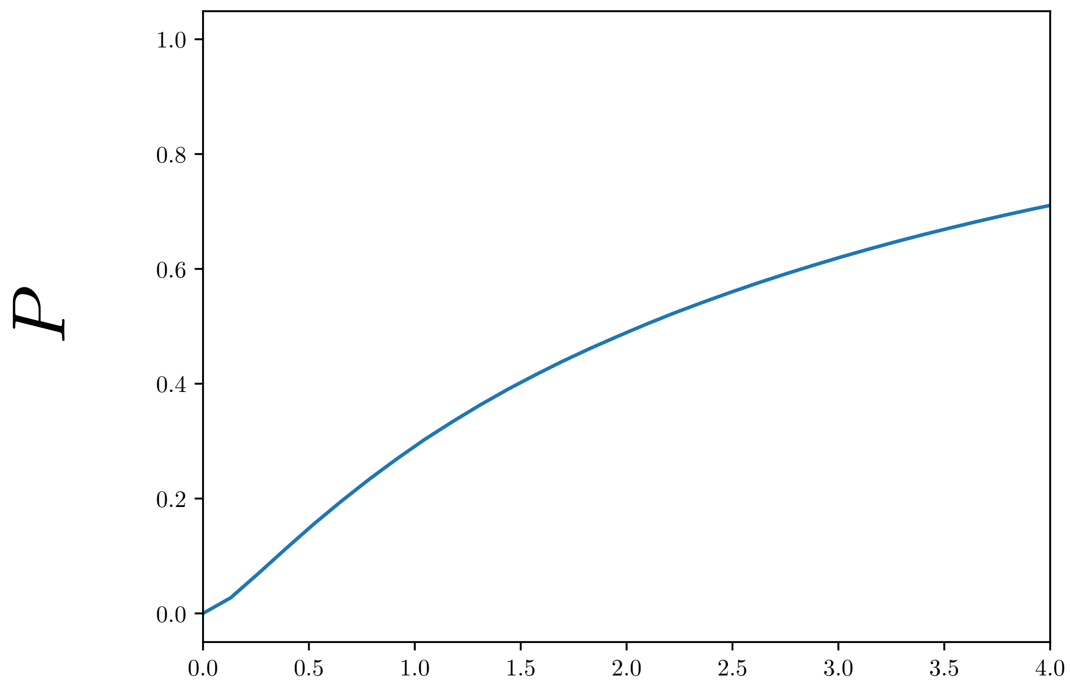


Protein interactions (C. Elegans)

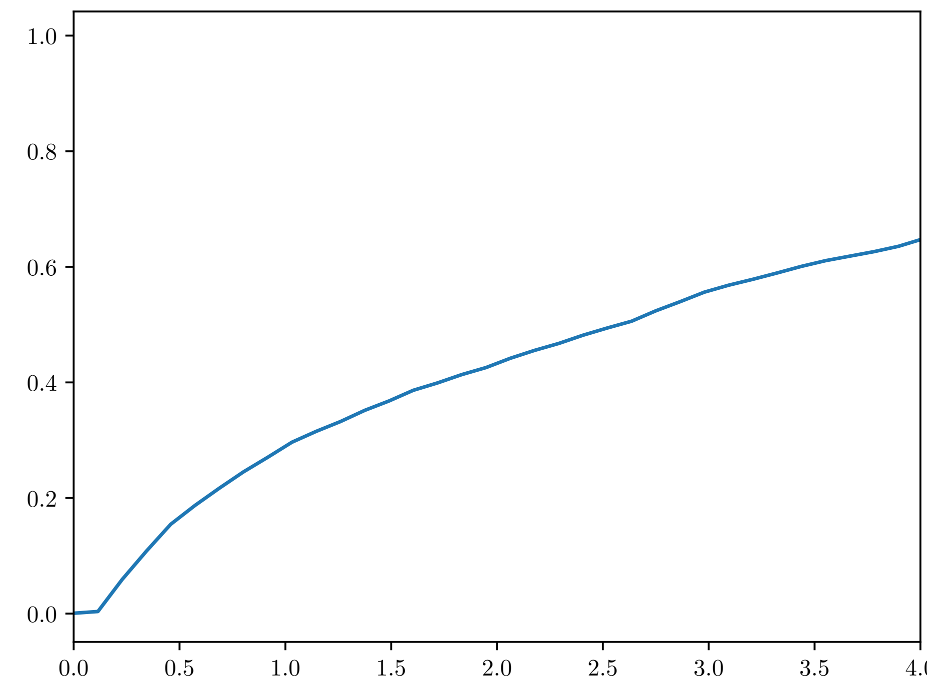


Rescaling the x-axis to use avg degree $\langle k \rangle = 2fL/N$

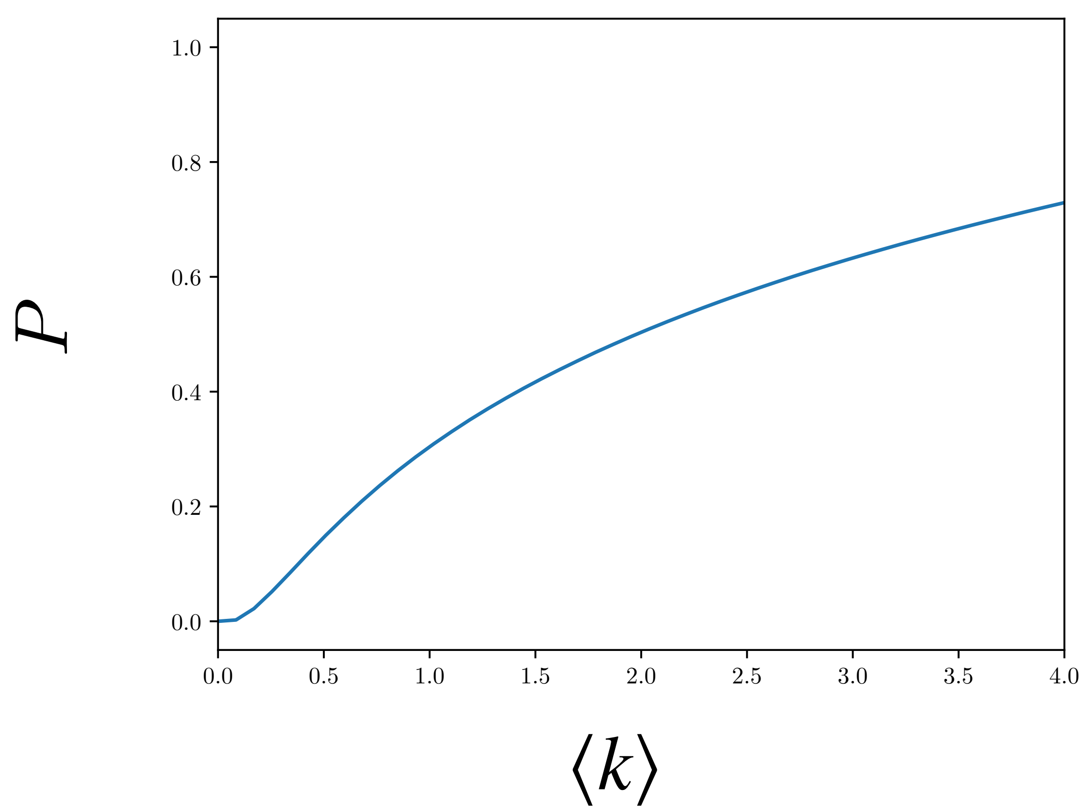
Internet autonomous systems



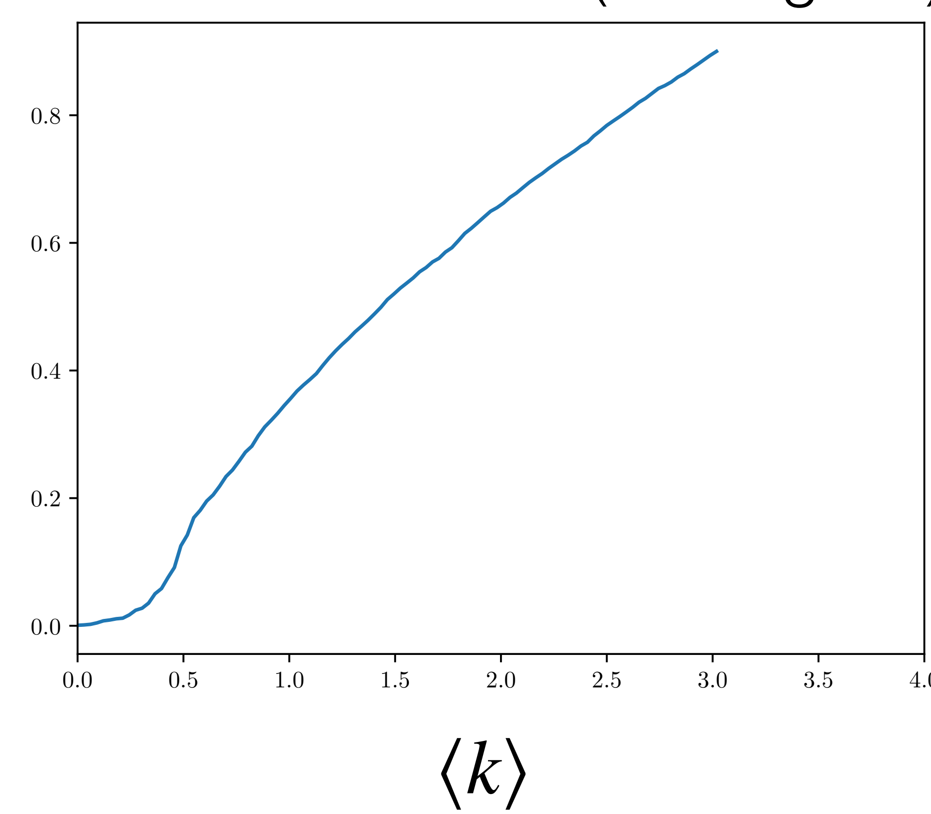
Flights between airports



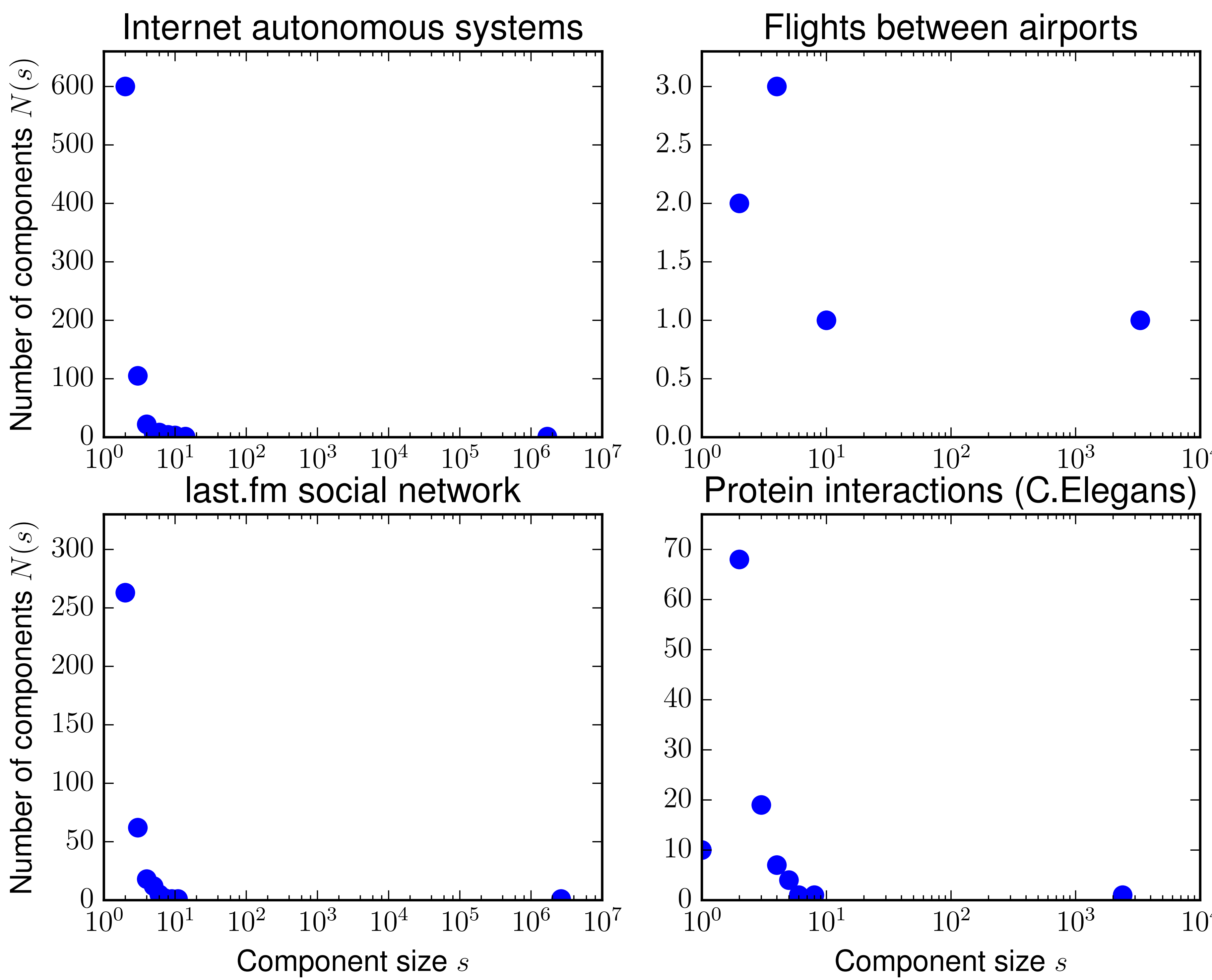
<u>last.fm</u> social network



Protein interactions (C. Elegans)



Components in data

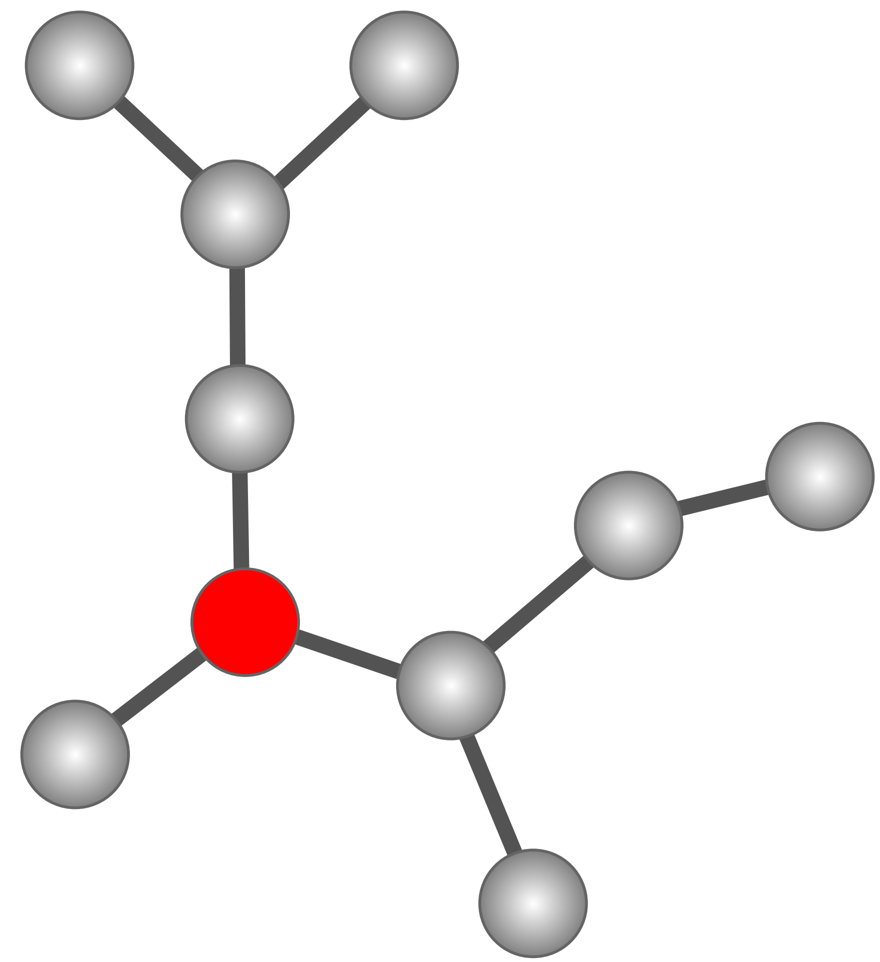


Summary so far

1. All models have a percolation transition
   - Disconnected phase, phase transition, connected phase
2. Transition points differ, sharpness varies
   - More variance in degrees = earlier transition
3. Curves for the data similar to the BA model

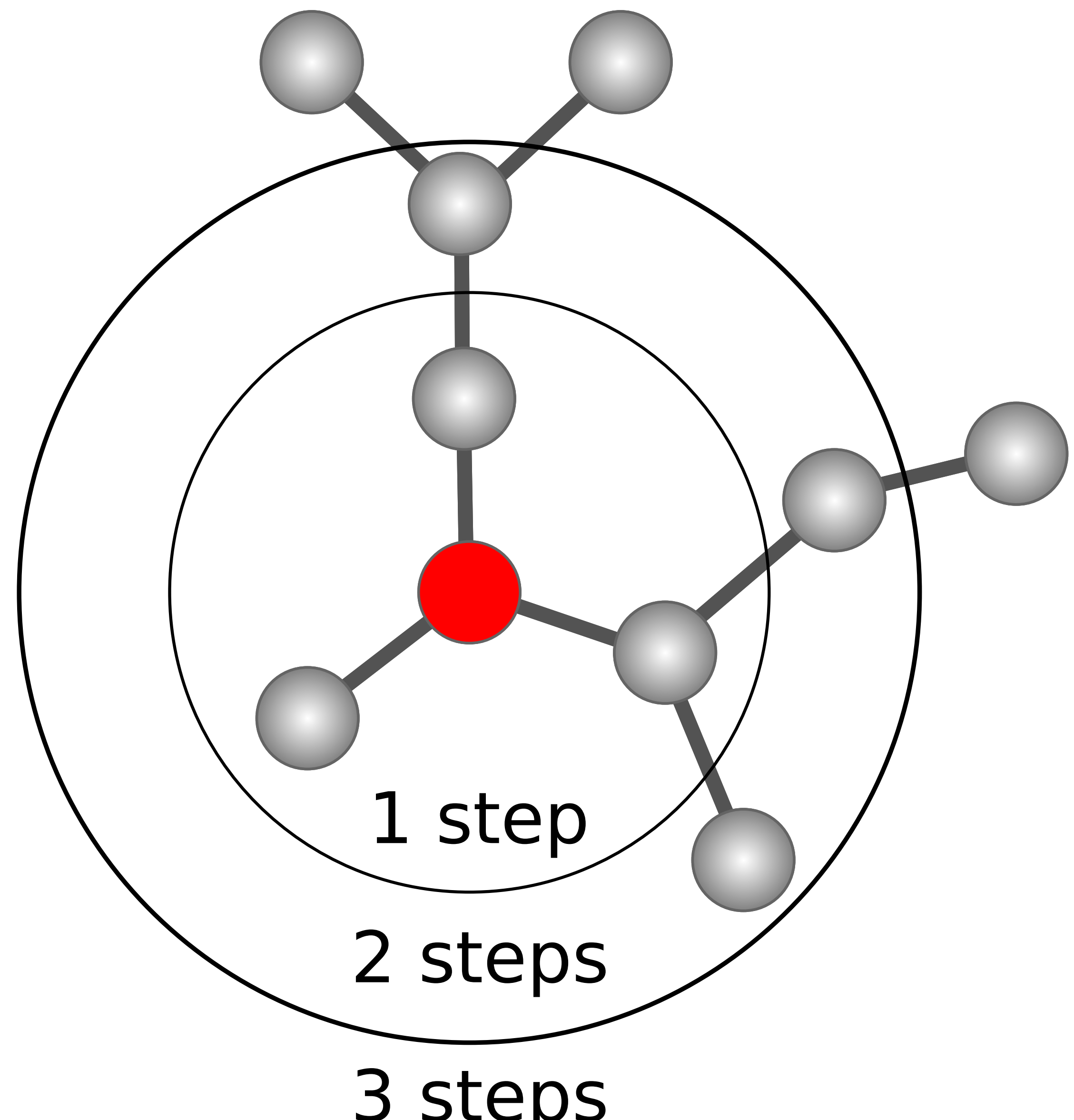
How to estimate the transition point?

- Idea: start from a random node, find how many nodes you can reach
- Before transition: you can always reach only a small number of nodes
- After transition: possibility of reaching very large number of nodes

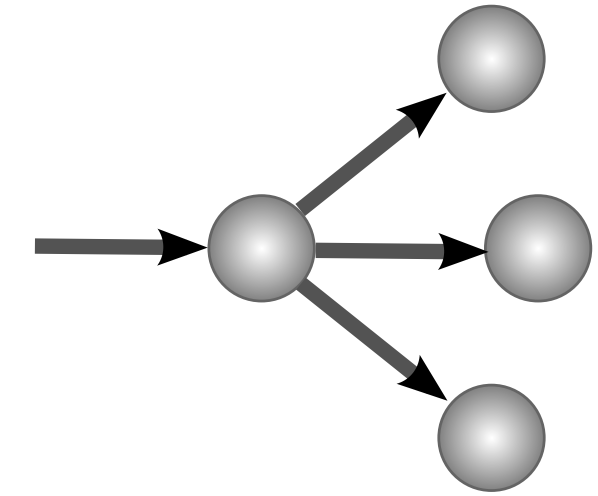


Branching processes

- Sparse large random networks have (almost) no loops
- Breadth first search is a "branching process":
  - A node has q "children"
- At step t, n_t nodes
  - $n_{t+1} = \langle q \rangle n_t$
  - Exponential growth ($\langle q \rangle > 1$) or decay ($\langle q \rangle < 1$)



Excess degree



- The excess degree q: *follow a link to a node, how many links does it have, not including the link that was followed?*
  - Remember the friendship paradox: following a link leads to high degree nodes: $\langle k_{nn} \rangle = \langle k^2 \rangle / \langle k \rangle$
- Expected excess degree: $\langle q \rangle = \langle k^2 \rangle / \langle k \rangle - 1$

expected number of neighbours

not including the link that was followed

How to estimate the transition point?

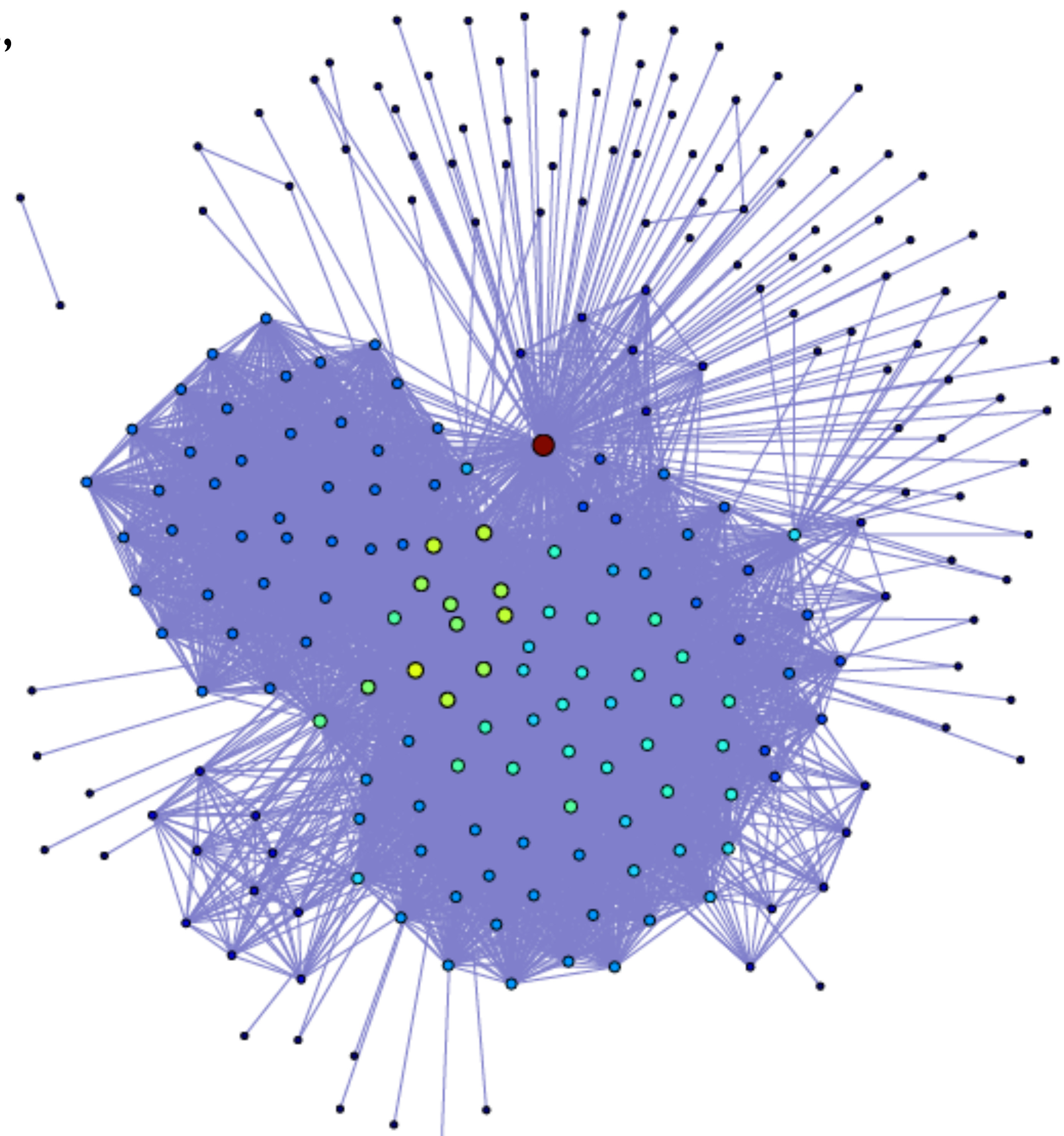
- From the branching process idea: $\langle q \rangle > 1$
- Expected excess degree: $\langle q \rangle = \langle k^2 \rangle / \langle k \rangle - 1$
- Giant component exits if $\boxed{\langle k^2 \rangle / \langle k \rangle > 2}$
  - For ER networks: $\langle k \rangle > 1$
  - High 2nd moment (fat tails) lead to high transition points
  - Remember: scale-free networks ($p(k) \propto k^{-\gamma}$) we got: $\langle k \rangle = \mathrm{const.}$ and $\langle k^2 \rangle = \infty$ if $2 < \gamma \leq 3$!

Different removal strategies

- Instead of edges, one can remove nodes (i.e., all links related to one node at a time)
- What if edges/nodes are not removed by random?
- What is the best removal strategy?
  - Attacking infrastructure networks
  - Breaking down the spreading networks of diseases (vaccinations etc.)

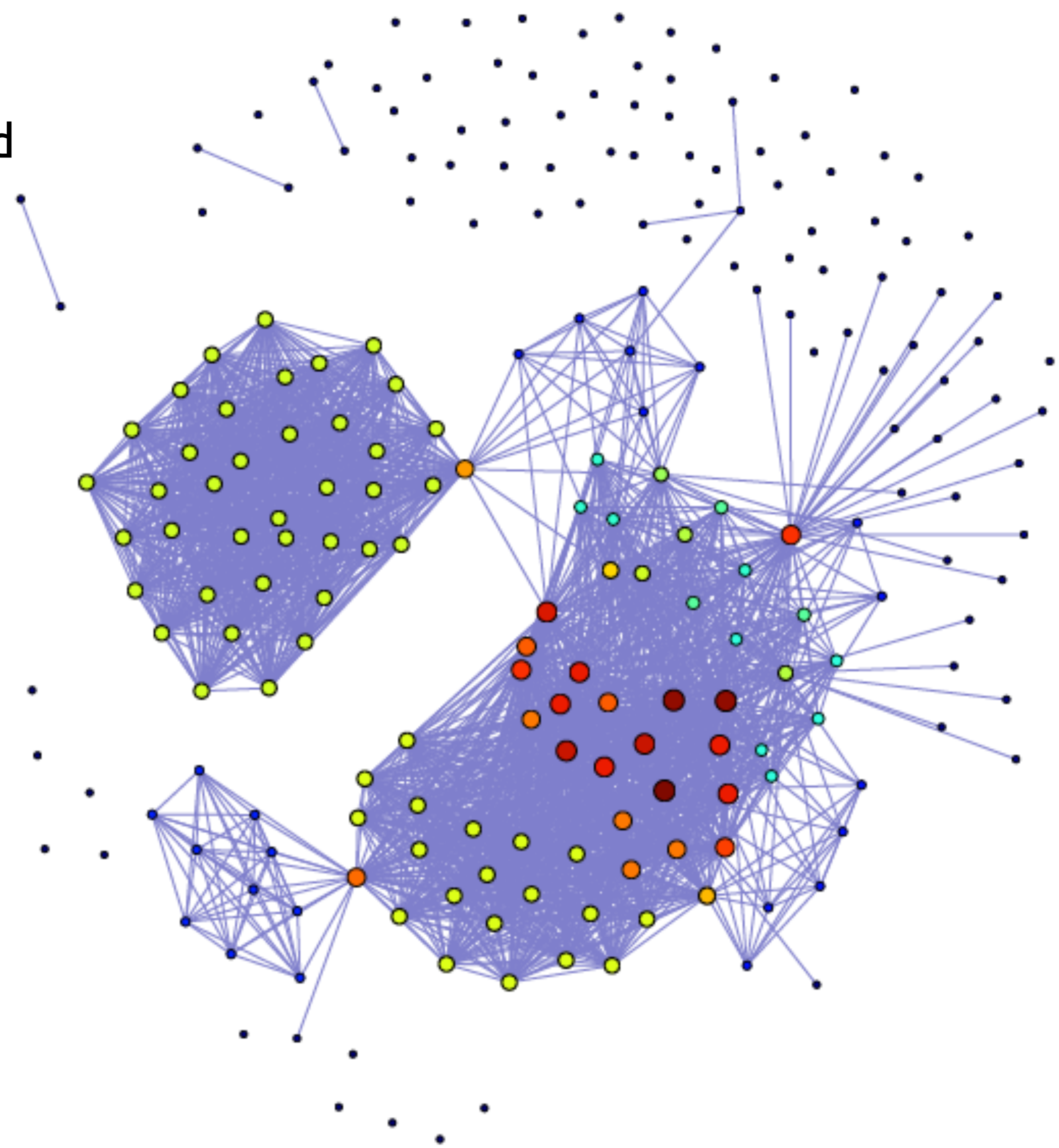
A simple example of the importance of hubs

- A project with workshops, meetings, etc
- Nodes are participants
- Nodes are linked if participants have co-attended meetings or workshops



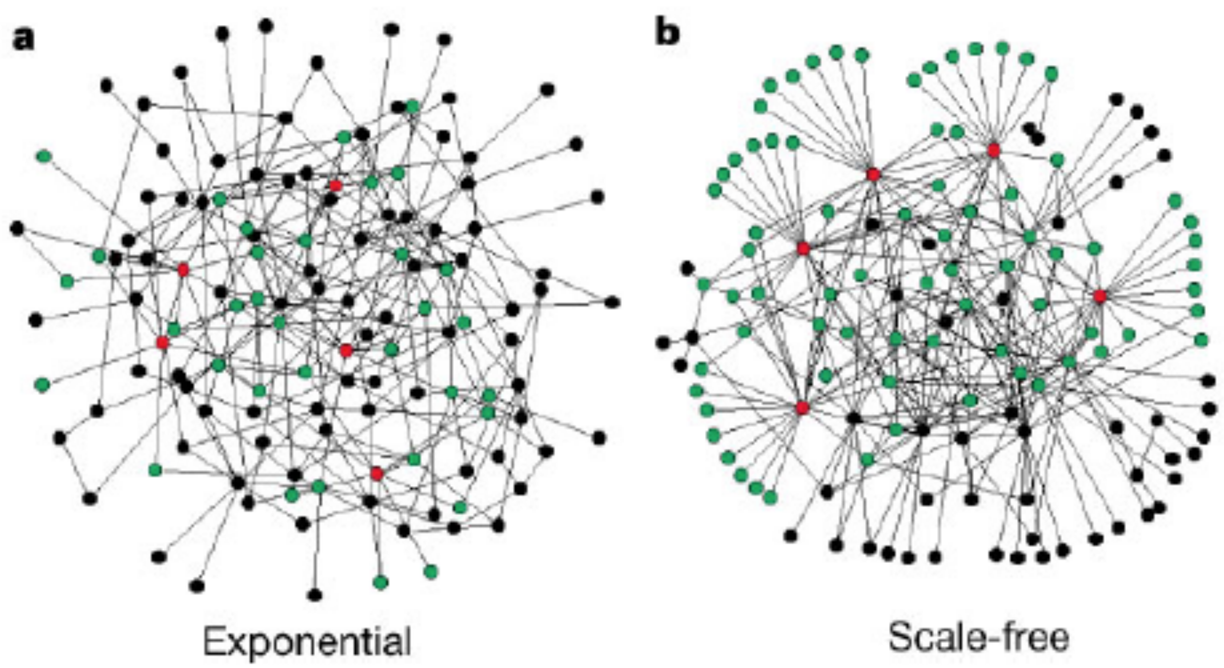
A simple example of the importance of hubs

- The same network, where the project coordinator and some leaders have been removed
- Part of the network fragments totally, other parts are less well connected
- This is not good, if the project is supposed to foster future collaborations and interaction between the participants!



Importance of hubs: network robustness

Albert, R., Jeong, H., & Barabási, A.-L., Nature 406, 378–383 (2000)



a Poisson random network

b Scale-free network

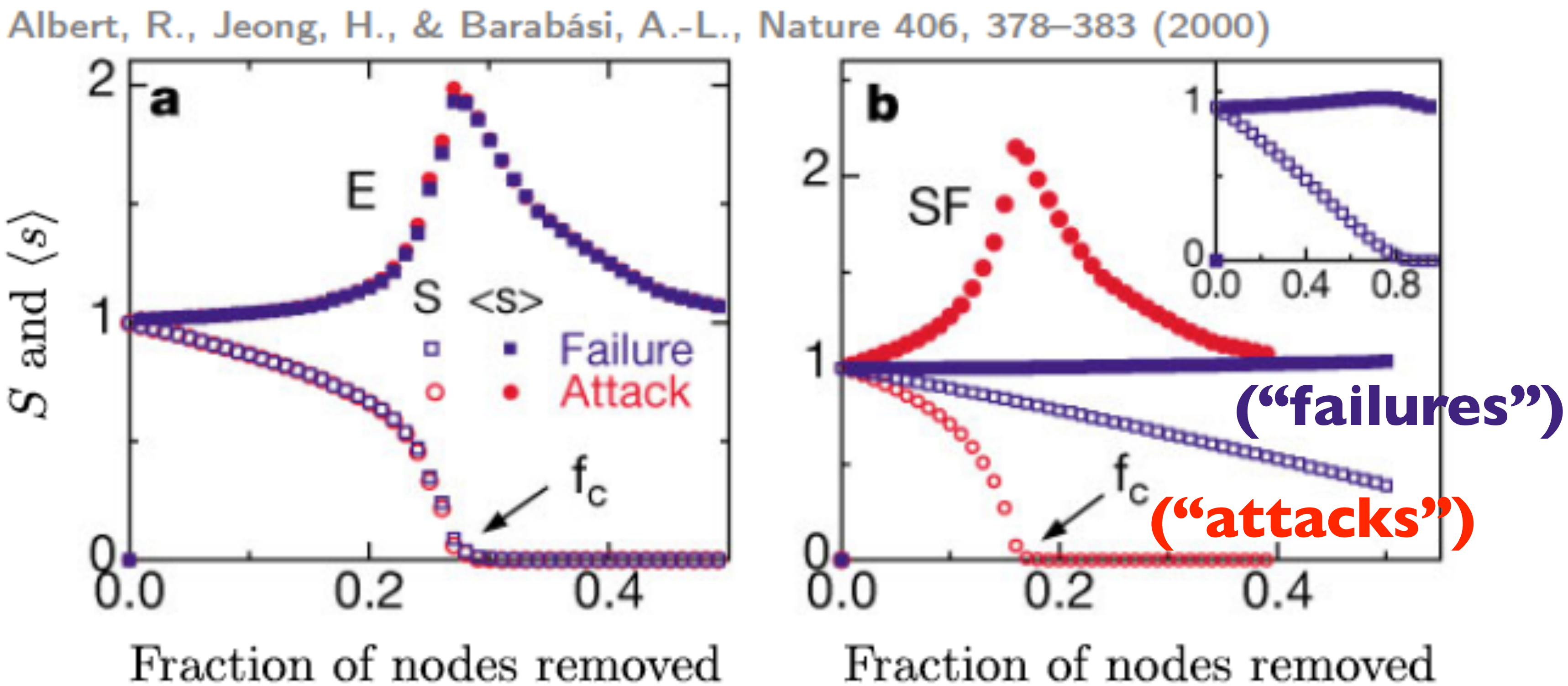
Both graphs have $n = 130$ and $\langle k \rangle = 3.3$.

Numerical experiment

1. Take a connected network
2. Remove vertices one at a time
3. Observe the size of the largest component

- Do the experiment so that
  1. Vertices are removed randomly **("errors")**
  2. Vertices are removed in descending order by degree (i.e. remove hubs first) **("attacks")**

Error and attack tolerance



S = relative size of the largest component

$\langle s \rangle$ = average size of other components

ER network

- Both methods of removal give the same result.
- The network breaks up ($S \to 0$) after a finite fraction of nodes is removed.

a Scale-free network

- Robust against random removal (in blue).
- Removing hubs first breaks up the network quickly.

SIR spreading model

- Each node in one of the three states
    - S : susceptible
    - I : infected
    - R : recovered
- Set one node to state I, rest in state S
- In the end, all nodes in S or R state

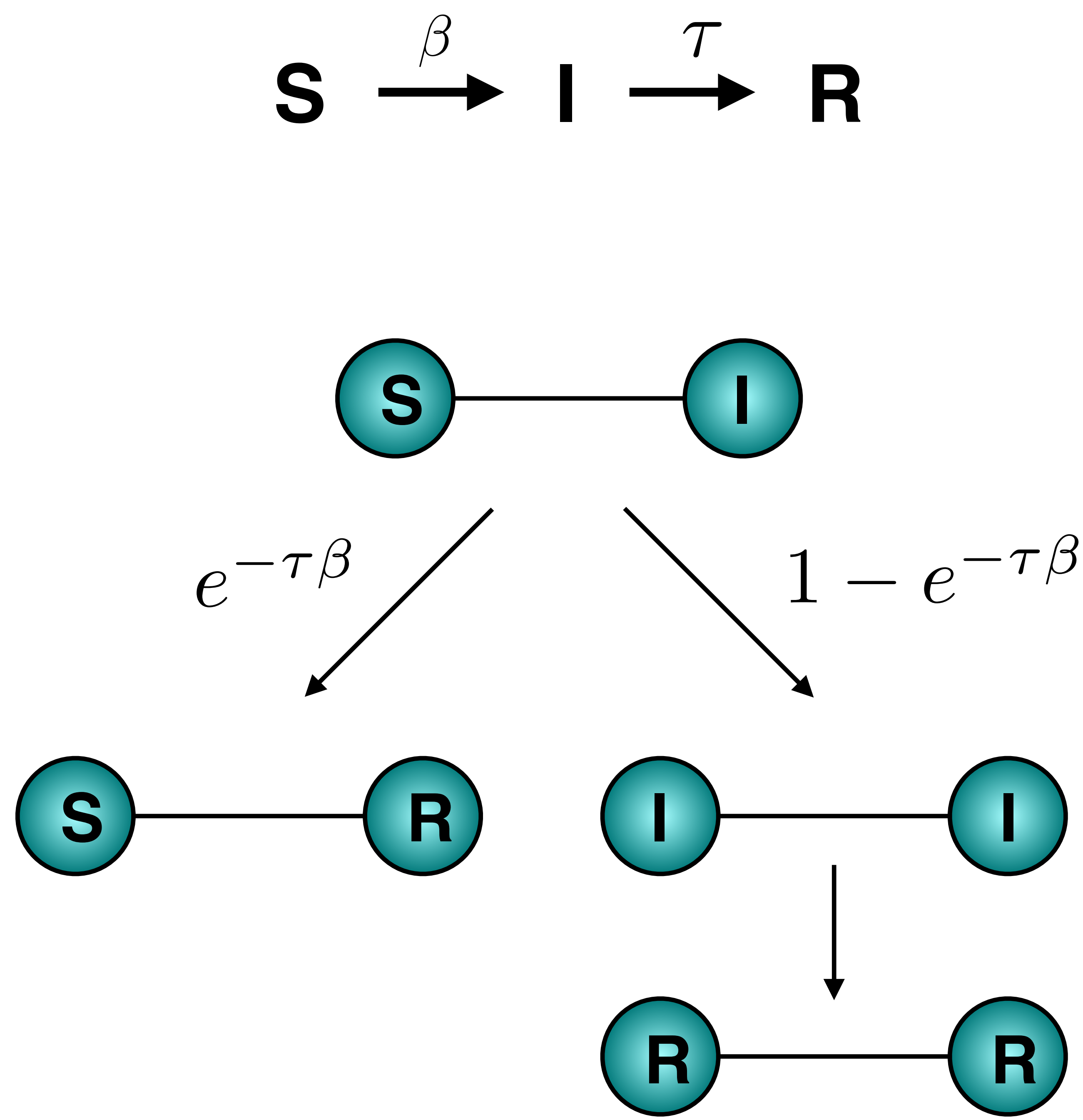
Infected nodes turn susceptible nodes they are connected to infected with rate β

Infected nodes remain infected for time τ

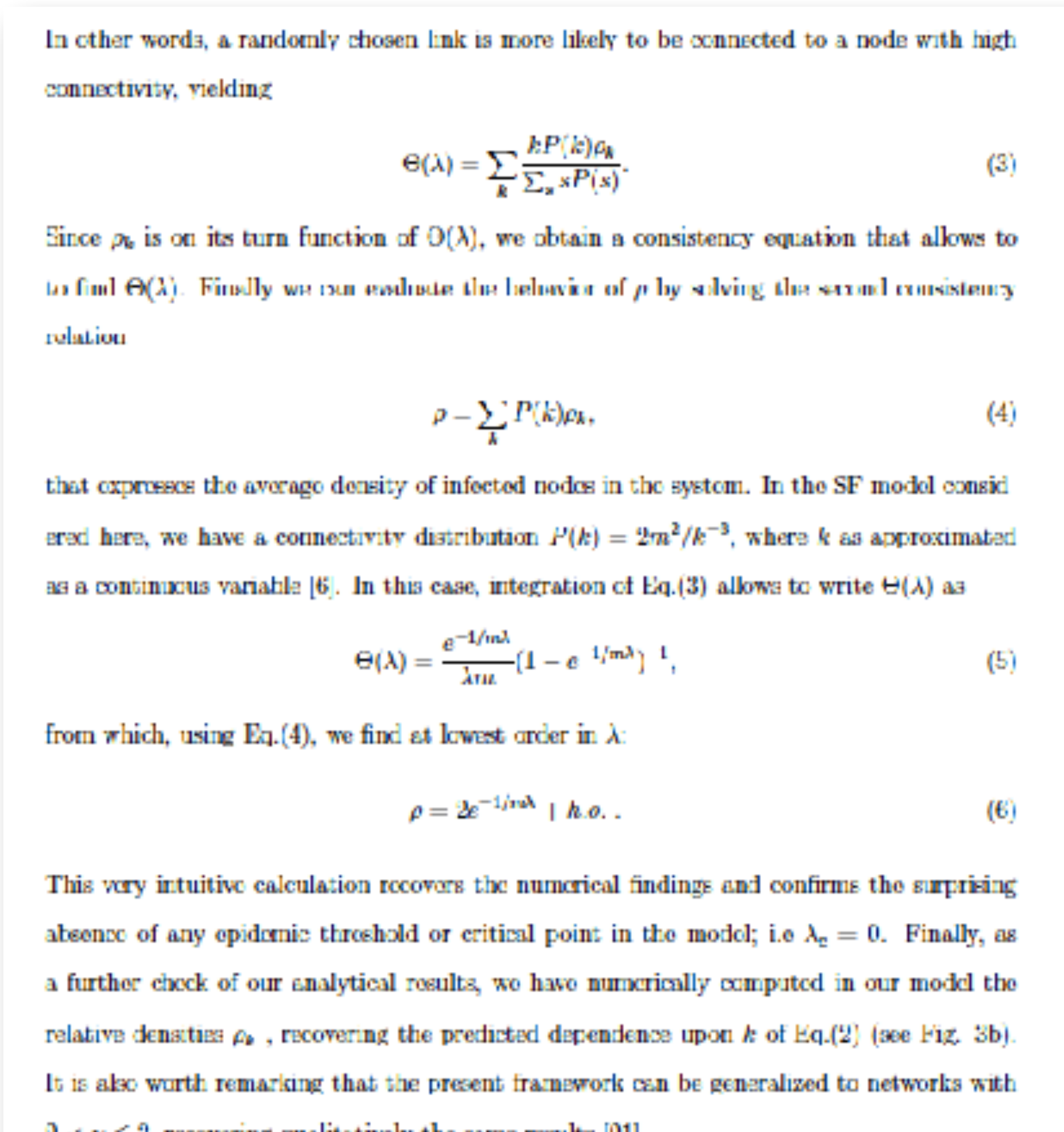
$$\mathbf{S} \xrightarrow{\beta} \mathbf{I} \xrightarrow{\tau} \mathbf{R}$$

SIR spreading model = percolation

- Finding how many nodes were infected in the end of the process can be mapped into a percolation problem!
- For a single link, the probability of transmitting infection is $\phi = 1 - e^{-\tau\beta}$
- Activate links with the probability ϕ
- Two nodes in a same component: infection can spread from one to the other
  - Fast way to calculate epidemic sizes distributions
  - Phase transition = epidemic appears



Scale-free networks have zero epidemic threshold!



In other words, a randomly chosen link is more likely to be connected to a node with high connectivity, yielding

$$\Theta(\lambda) = \sum_k \frac{kP(k)\rho_k}{\sum_s sP(s)}. \tag{3}$$

Since ρ_k is on its turn function of $\Theta(\lambda)$, we obtain a consistency equation that allows to to find $\Theta(\lambda)$. Finally we can evaluate the behavior of ρ by solving the second consistency relation

$$\rho = \sum_k P(k)\rho_k, \tag{4}$$

that expresses the average density of infected nodes in the system. In the SF model considered here, we have a connectivity distribution $P(k) = 2m^2/k^{-3}$, where k as approximated as a continuous variable [6]. In this case, integration of Eq.(3) allows to write $\Theta(\lambda)$ as

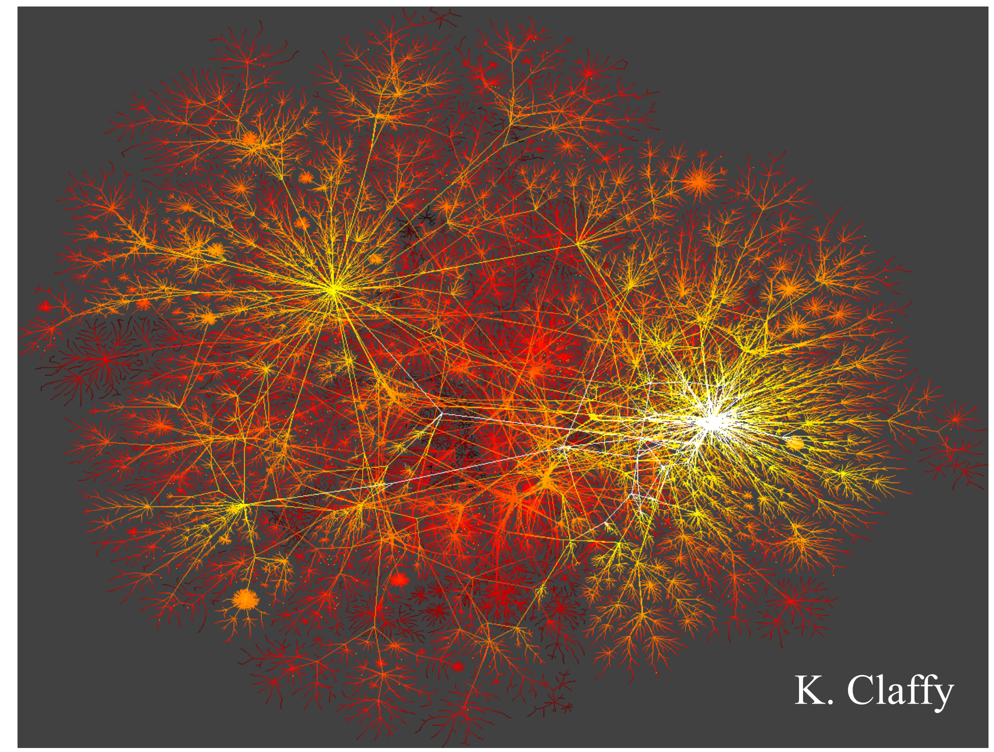
$$\Theta(\lambda) = \frac{e^{-1/m\lambda}}{\lambda m}(1 - e^{-1/m\lambda})^{-1}, \tag{5}$$

from which, using Eq.(4), we find at lowest order in λ:

$$\rho = 2e^{-1/m\lambda} + h.o.. \tag{6}$$

This very intuitive calculation recovers the numerical findings and confirms the surprising absence of any epidemic threshold or critical point in the model; i.e $\lambda_c = 0$. Finally, as a further check of our analytical results, we have numerically computed in our model the relative densities ρ_k, recovering the predicted dependence upon k of Eq.(2) (see Fig. 3b). It is also worth remarking that the present framework can be generalized to networks with $2 < \gamma \leq 3$, recovering qualitatively the same results [21].

R. Pastor-Satorras and A. Vespignani (2001). "Epidemic spreading in scale-free networks". *Physical Review Letters* **86** (14): 3200–3203



...which partially explains why computer viruses are hard to eradicate...

The importance of hubs

- These results indicate that **real-world networks are robust to random failures**
  - e.g. however many servers are down, the Internet still exists!
- This has its price: **targeted attacks rapidly break down networks**
- This has consequences e.g. on the spreading of biological and electronical viruses (random removal of nodes by vaccination won't help!)
- Thus to mitigate viral spreading, one has to protect the hubs!

Want to know more?

CS-E5745 Mathematical Methods in Network Science (*period III*). Topics include:

- More detailed theory how to calculate component size distributions, percolation thresholds/critical points in random networks
- Using the same tools to estimate theoretical epidemic thresholds in random networks
- … and other topics