Large-scale structures in networks: Hidden communities and latent hierarchies

Daniel Larremore
Dept. of Computer Science
& BioFrontiers Institute

15 August 2018
SICSS

University of Colorado **Boulder**

daniel.larremore@colorado.edu
@danlarremore

Goals for this talk:

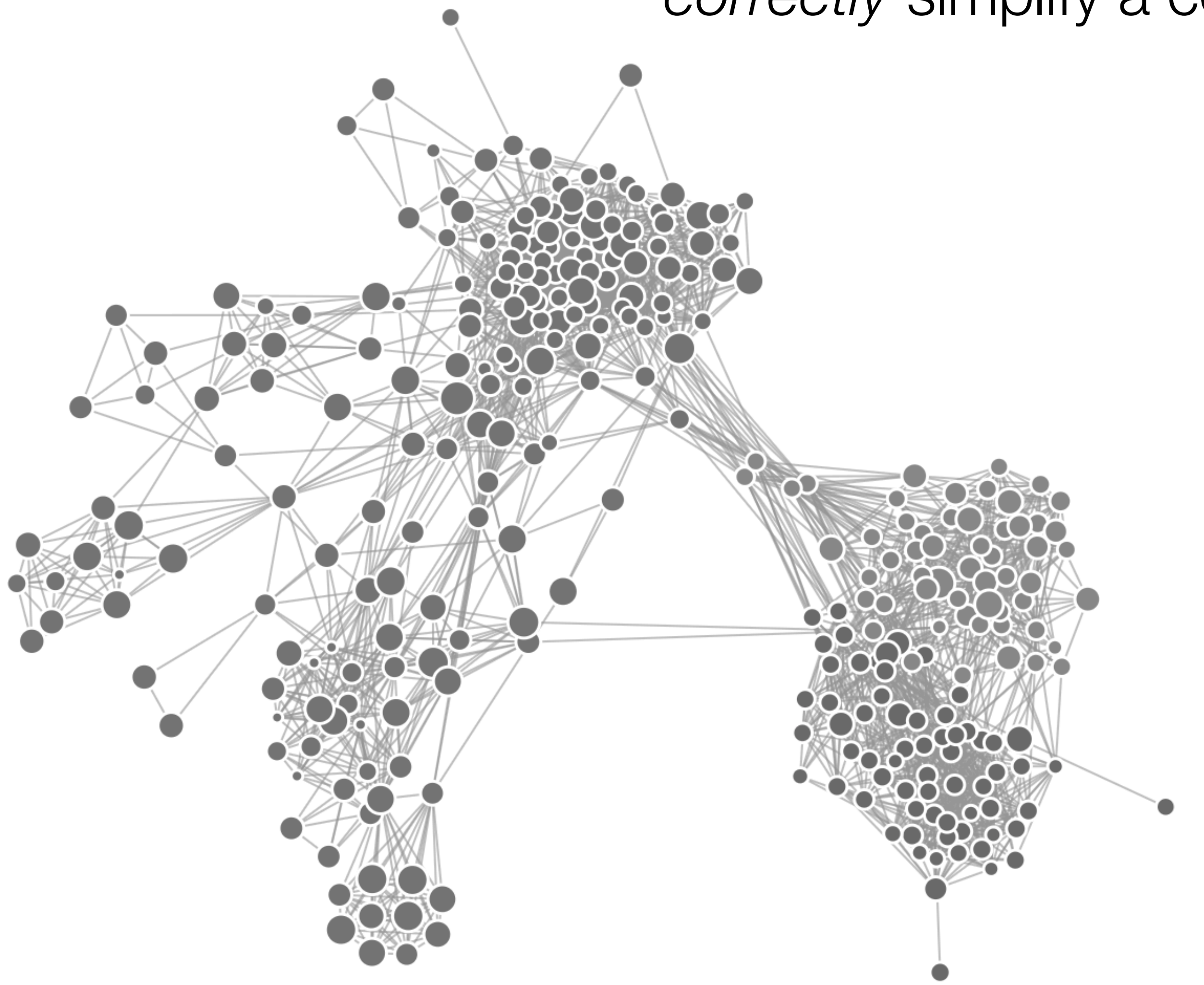
1. **Why** do we look for large-scale structure? 🤔
2. **How** do we find communities and hierarchies? 🧐
3. **Where** can we read more details? 📚

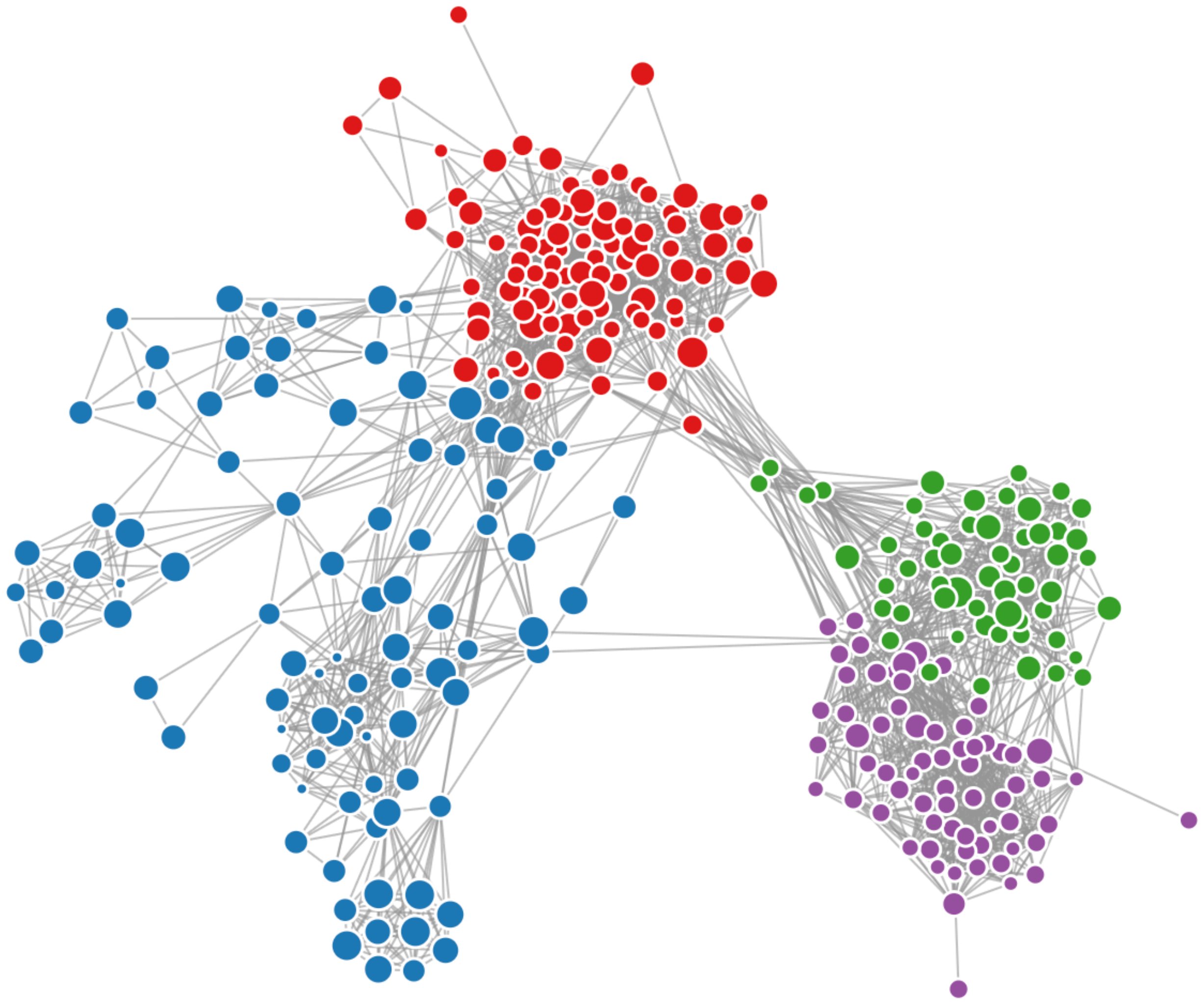
Simplicity is a great virtue but it requires hard work to achieve it and education to appreciate it. And to make matters worse: complexity sells better.
E. W. Dijkstra

We can interpret this in two ways:

The Cynic: Pictures of networks can be *really cool* but our goal is to do good science, not make pretty pictures.

The Scientist: The most beautiful science is when we *correctly* simplify a complex system.





What do we mean by “large-scale structure” ?

Structure is what makes data different from noise.
It’s what makes a network different from a random graph.

Networks are often too large and complex to be adequately summarized by a few scalars, like the number of nodes, the number of edges, or the mean degree.

However, they are also often too large and complex to be analyzed *without* some kind of simplification!

Therefore, understanding what the network *means* requires that we identify key structures.

Searching for large-scale structures in a network reflects a belief that in all the complexity there are patterns that make the network less complicated.

We define these large-scale structures—models, really—to compress complex networks.

Goal: understanding, not a list of parts and dimensions

Finding large-scale structures
is the same as anything else:

We want a **simplified model** of
something very complicated.

We want to know what the
important pieces are,
and **how they fit together**.

Many uses for models of large-scale structure

Treat the network like a system:

Extrapolation. Make predictions for as-yet unseen nodes (in “space” or time).
Interpolation. Identify missing links.
Generalization. Nodes of this type are like others of the same type.

Treat the network like an artifact:

Mechanisms. How did this network arise? What rules governed its assembly?
Explanations. Coarse-graining or compression.

Treat the network like a means to an end; an intermediate data structure:

Useful division. Need groups so that we can assign treatments in an A/B test.
Simplification. Downstream regression model needs ranks or groups.

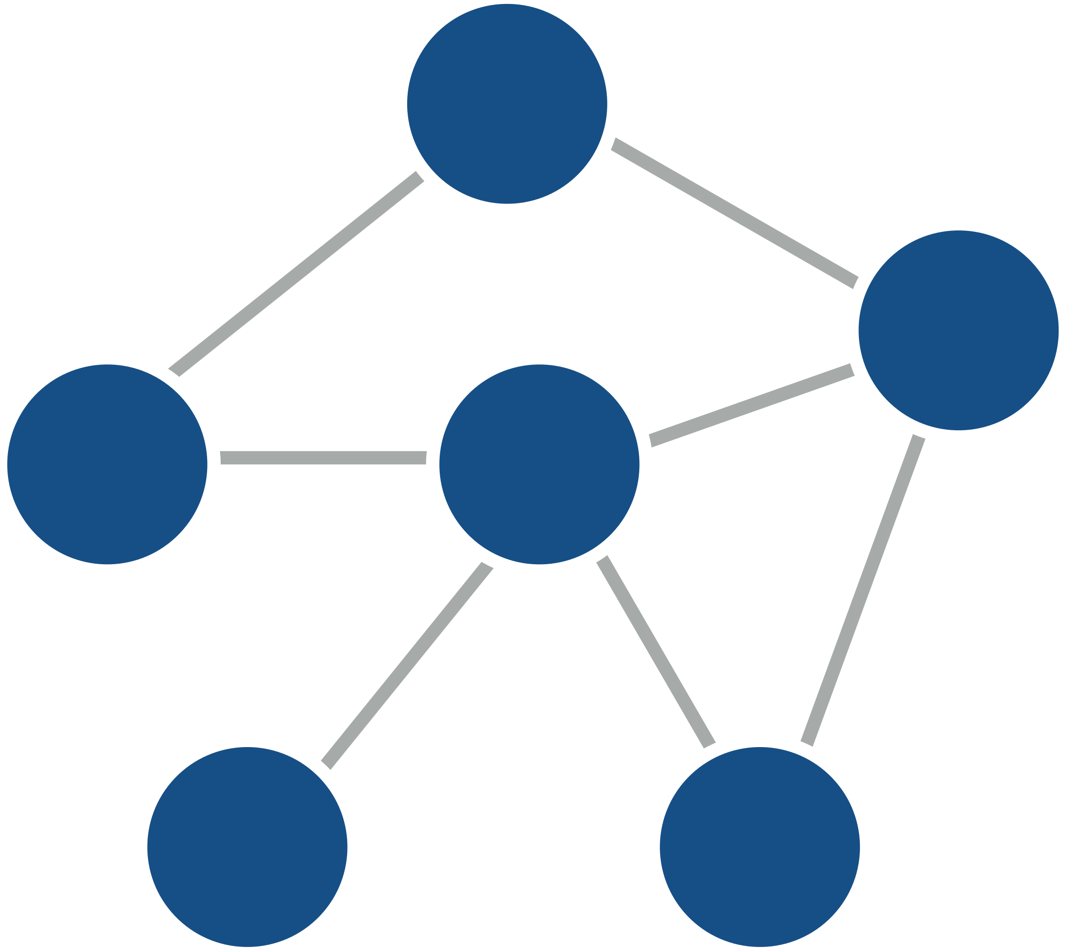
intuition: compare this list with the list you would write for regression

Community structure

Homophily & assortative mixing

like links with *like*

Assortativity coefficient *r* measures extent of homophily.



Newman, Phys. Rev. E 67, 026126 (2003).

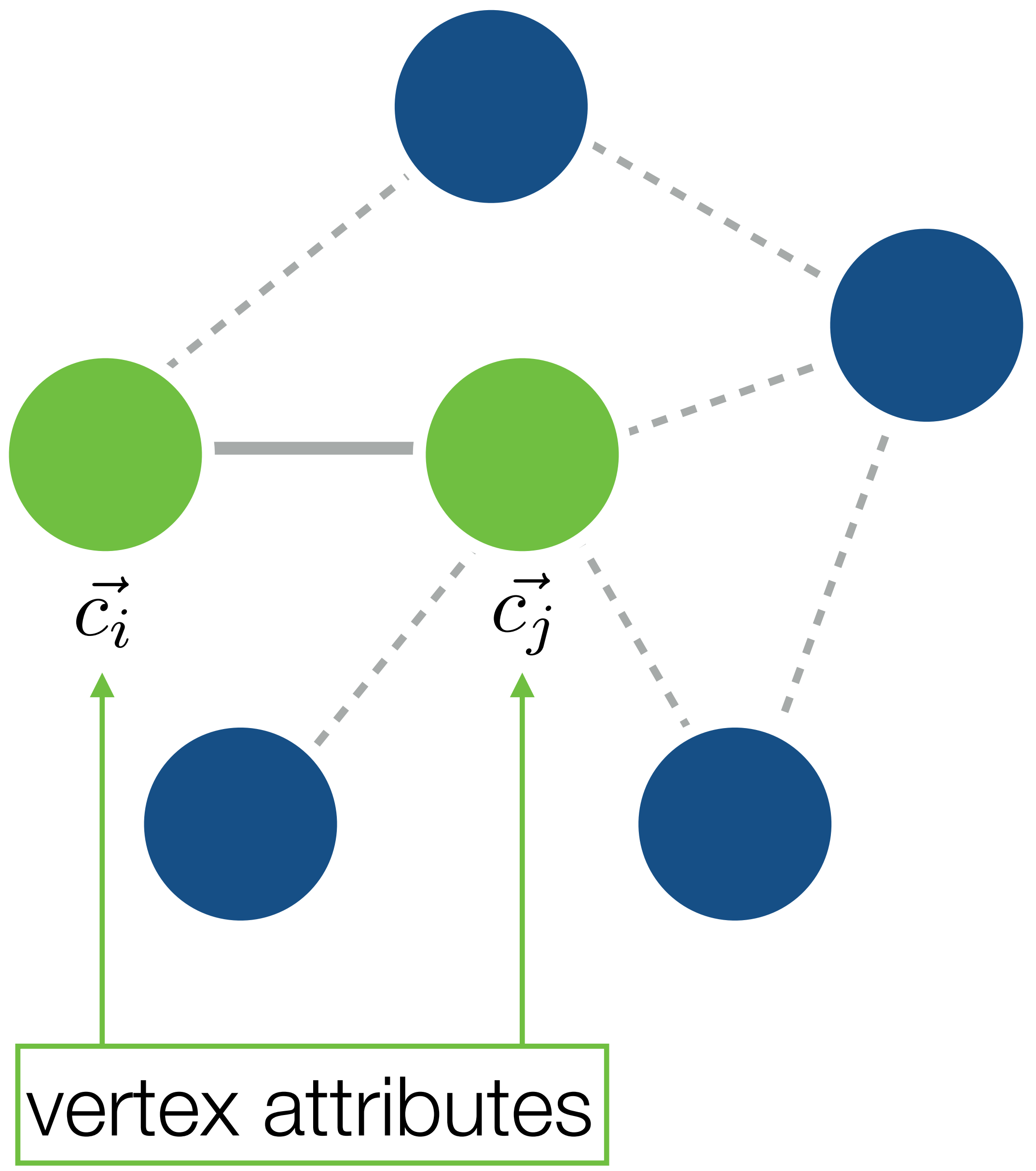
Homophily & assortative mixing

like links with *like*

Assortativity coefficient measures extent of homophily.

Three types:

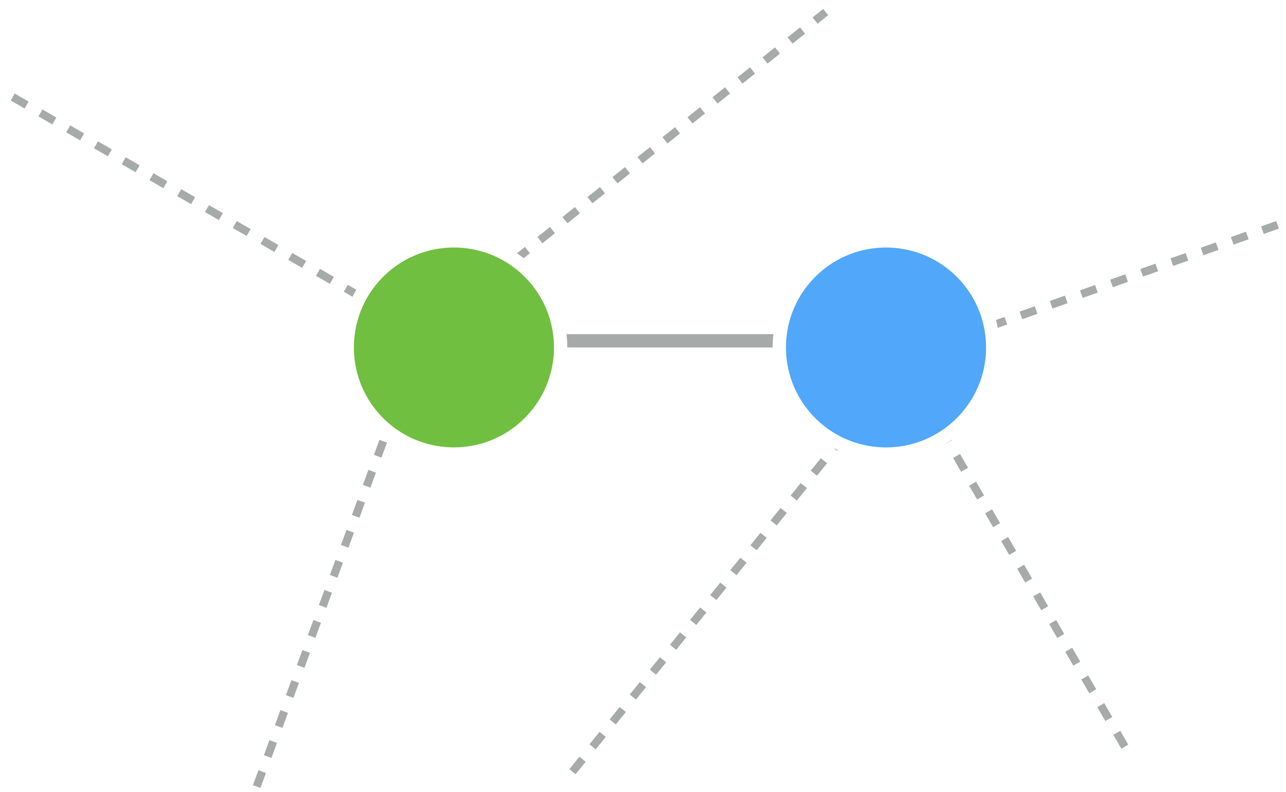
- scalar attributes
- vertex degrees
- categorical variables



Newman, Phys. Rev. E 67, 026126 (2003).

Homophily & assortative mixing

like links with *like*



It is convenient to write the correlation of categories across edges this way, and call it *Q.*

Principle: what fraction of edges fall between nodes of the same community?*

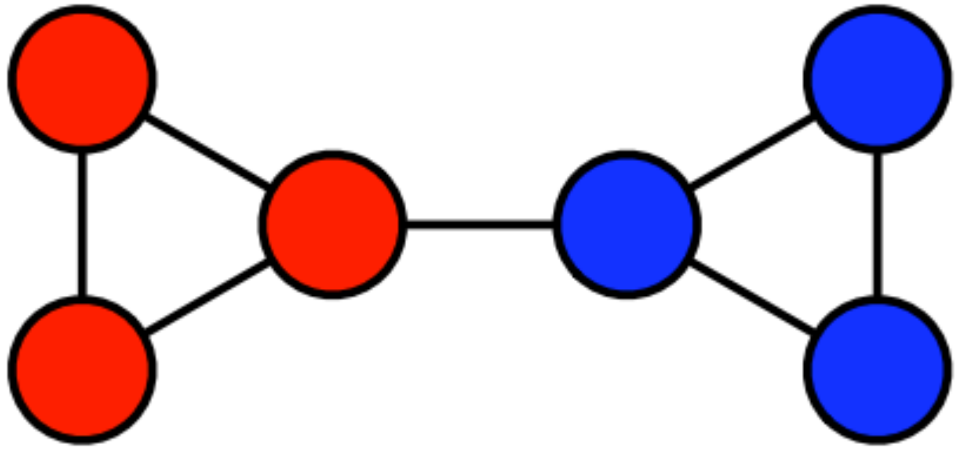
$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta_{ij}$$

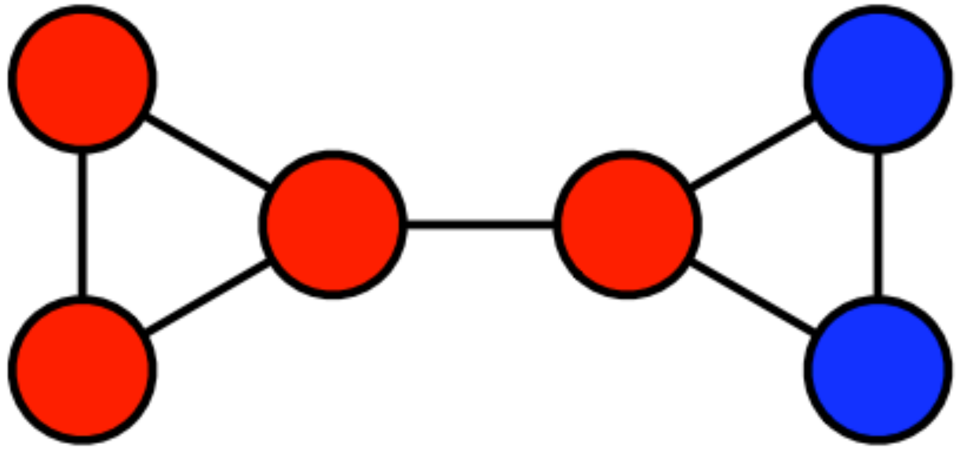
*compared to what we'd expect if the network were random!

Practice makes the master

$$Q = \frac{1}{2m}\sum_{ij}\left(A_{ij} - \frac{k_i k_j}{2m}\right)\delta_{b_i,b_j}$$

$$Q = \sum_r e_{rr} - a_r^2$$





labeling 1	red	blue
red	3/7	1/14
blue	1/14	3/7

$Q_1 = 5/14 = 0.357$

labeling 2	red	blue
red	4/7	2/14
blue	2/14	1/7

$Q = 6/49 = 0.122$

Modularity

Modularity is easily *the* most popular method for community detection. But why?

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta_{b_i, b_j}$$

Why is this more powerful than simply a measure of correlation over node labels?


Community structure in social and biological networks
M Girvan, MEJ Newman - Proceedings of the national …, 2002 - National Acad Sciences
Abstract A number of recent studies have focused on the statistical properties of networked systems such as social networks and the Worldwide Web. Researchers have concentrated particularly on a few properties that seem to be common to many networks: the small-world
Cited by 10319 ted articles All 54 versions

Finding and evaluating community structure in networks
MEJ Newman, M Girvan - Physical review E, 2004 - APS
Abstract We propose and study a set of algorithms for discovering community structure in networks—natural divisions of network nodes into densely connected subgroups. Our algorithms all share two definitive features: first, they involve iterative removal of edges from
Cited by 9247 ted articles All 44 versions

Girvan & Newman, 2002. Community structure in social and biological networks. PNAS 99, 2002.

Key: let's reverse our thinking of what Q does

Don't use Q to compute correlation of some given labels.

Instead, **experiment with the labels** and see how you can **maximize Q**!

Now, we have a computer science problem:
how do you search the space of partitions?

(This space is really big!)

How would *you* do it? 🧐

People like modularity. Why?

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta_{b_i, b_j}$$

- Intuitive
- Works for weighted and unweighted networks.
- Corresponds to our social network ideas of what (cohesive) communities are.
  - Automatically choose *k,* the number of groups.
  - Rapid approximate solutions.
  - Follows the usual methods trajectory: idea, demonstration, optimization.

  - Fun customizations:
    - Resolution parameter to “zoom in” and “zoom out.”
    - Find the clusters. Then cluster the clusters. Then cluster those clusters…
    - Directed. Bipartite.

$$Q = \frac{1}{m} \sum_{ij} \left(A_{ij} - \frac{k_i^{\text{out}} k_j^{\text{in}}}{m} \right) \delta_{b_i, b_j}$$

modularity for directed networks

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \gamma \frac{k_i k_j}{2m} \right) \delta_{b_i, b_j}$$

modularity with a resolution parameter

Why aren’t we done here?

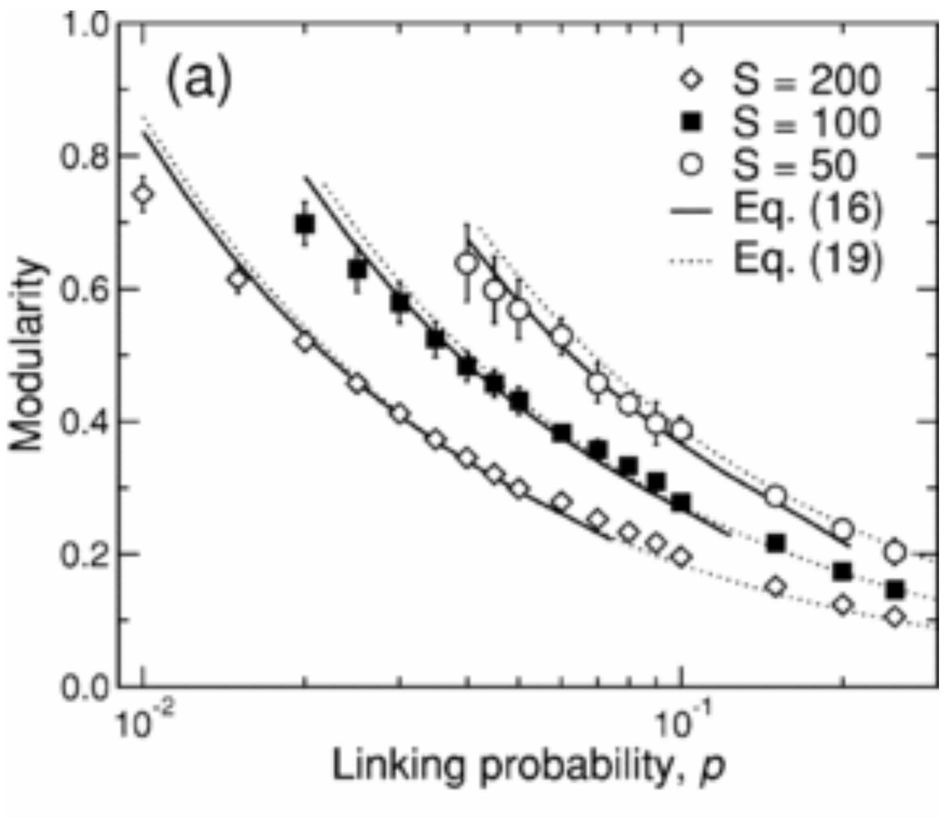
Physicists like to minimize things because rocks fall.
— Cris Moore

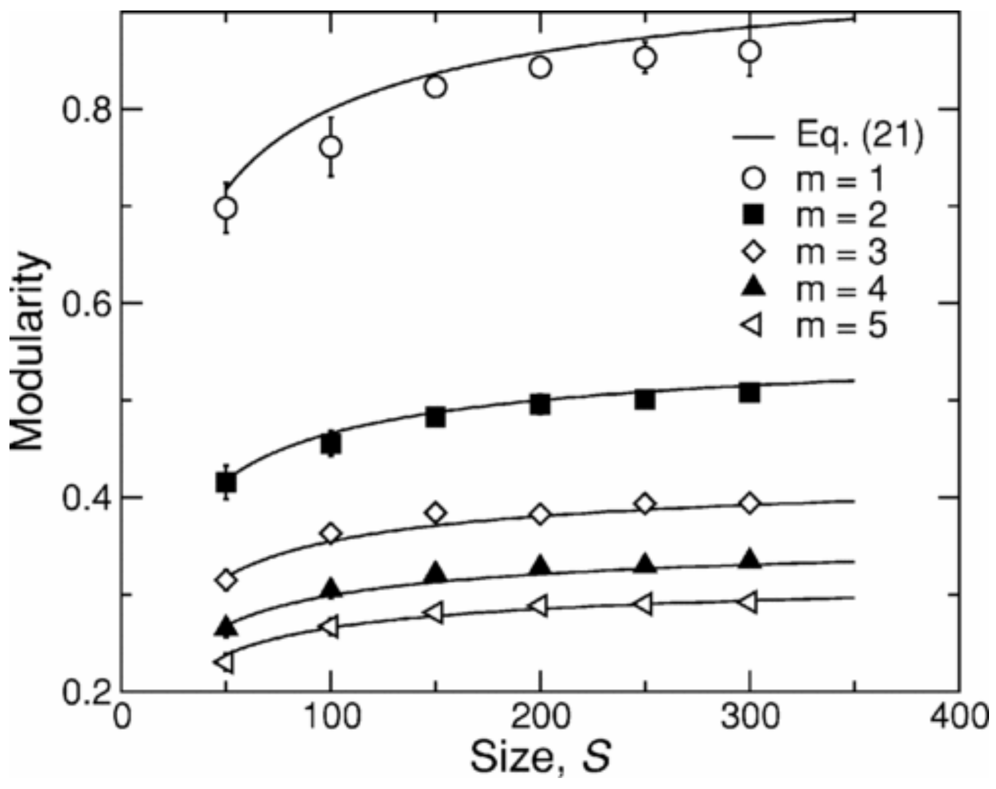
We can always maximize Q to find a partition, but is it meaningful?

Fooled by "structure" in totally random networks

As it turns out, you can find **high-modularity partitions in random networks**.

Structure is what makes data different from noise.
It's what makes a network different from a random graph.





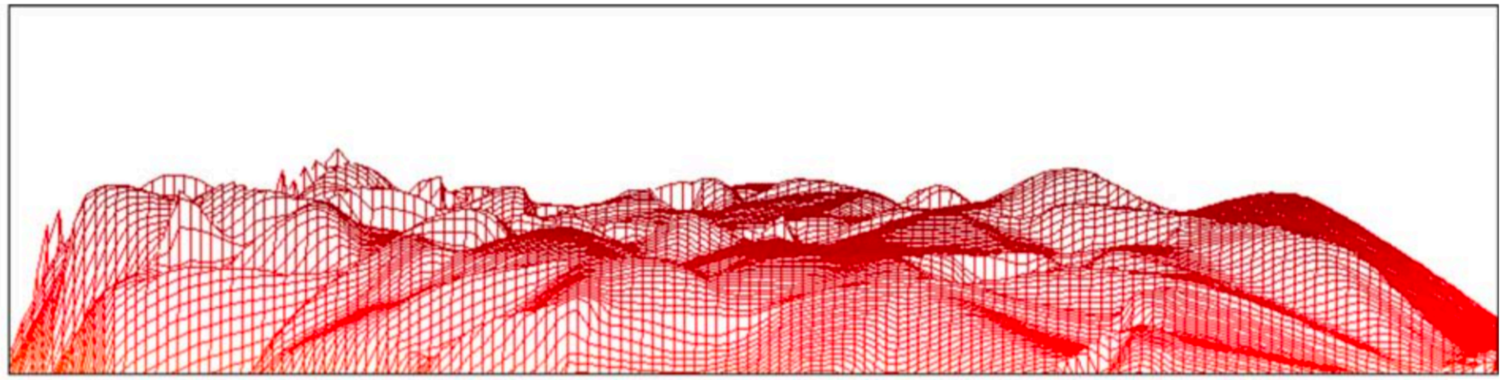
We prefer that our methods fail gracefully, and tell us when they fail. (like R^2)

[alternative perspective: maybe you *want* to find clusters in randomness?]

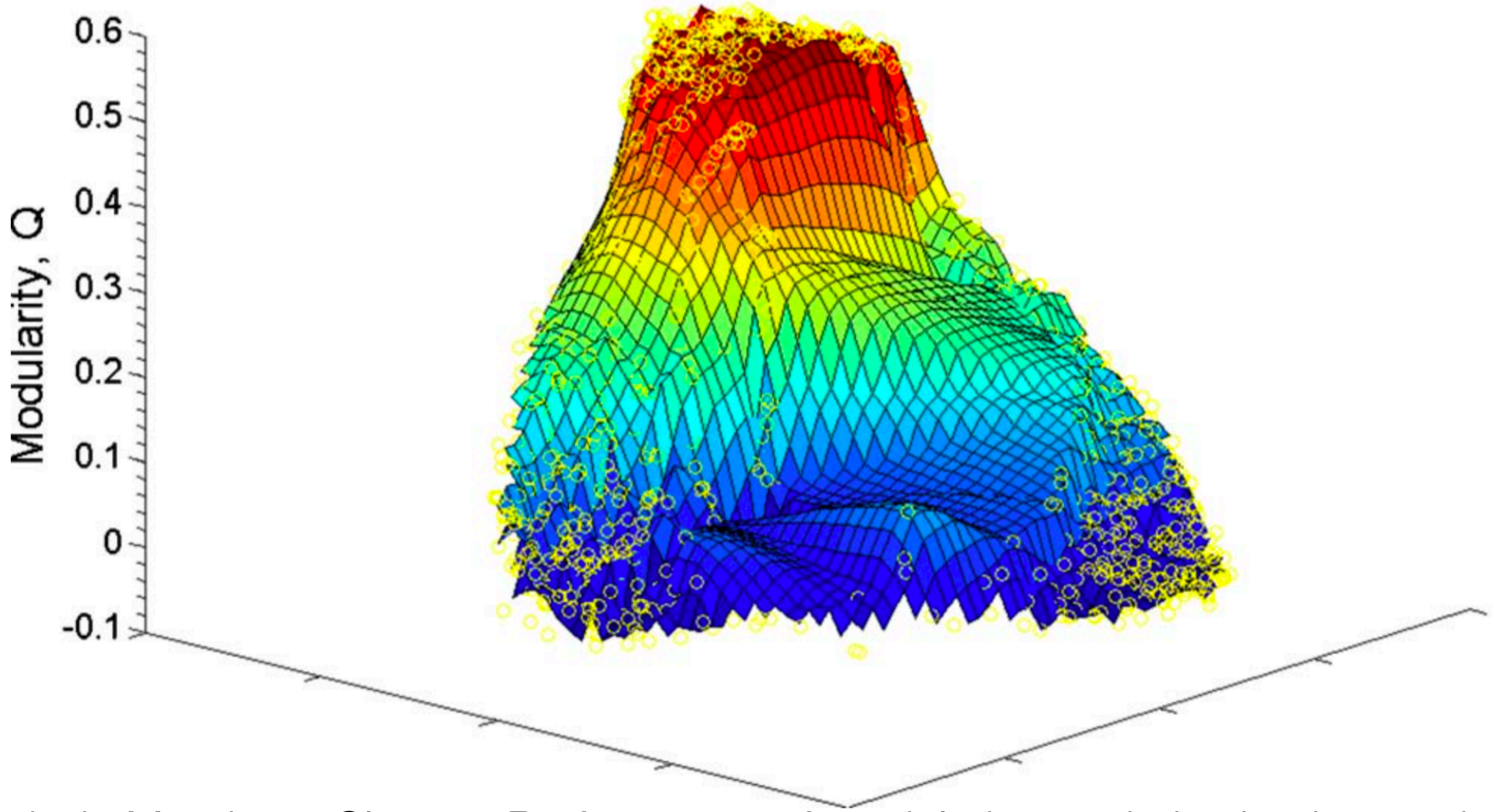
Guimera, Sales-Pardo, Amaral. "Modularity from fluctuations in random graphs and complex networks." Physical Review E 70.2 (2004): 025101.

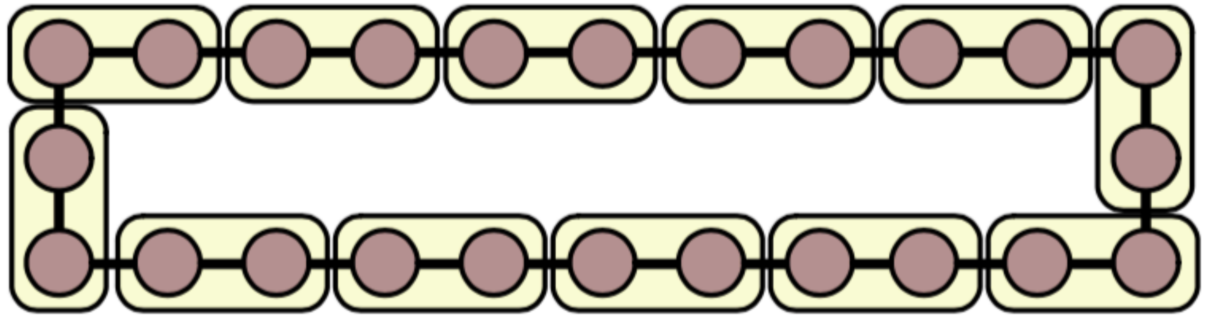
Modularity: degeneracy and strange behavior

Lots of different but nearly-as-good partitions.
The optimization landscape is *degenerate*.



Unintuitive behavior: find the communities in a chain of cliques and you get pairs of cliques…not the cliques themselves!

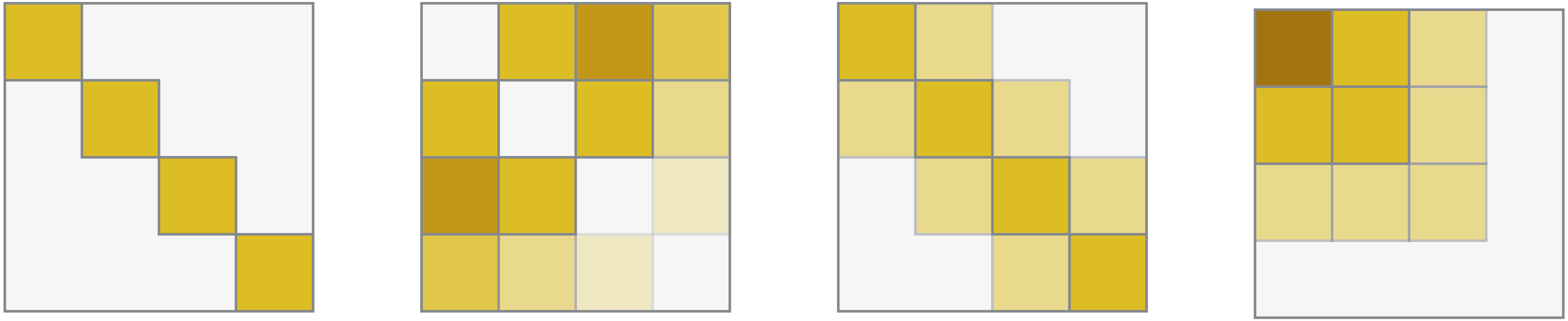




Good, de Montjoye, Clauset. Performance of modularity maximization in practical contexts. Phys. Rev. E 81, 046106 (2010).

Q is restricted to *assortative* community structure

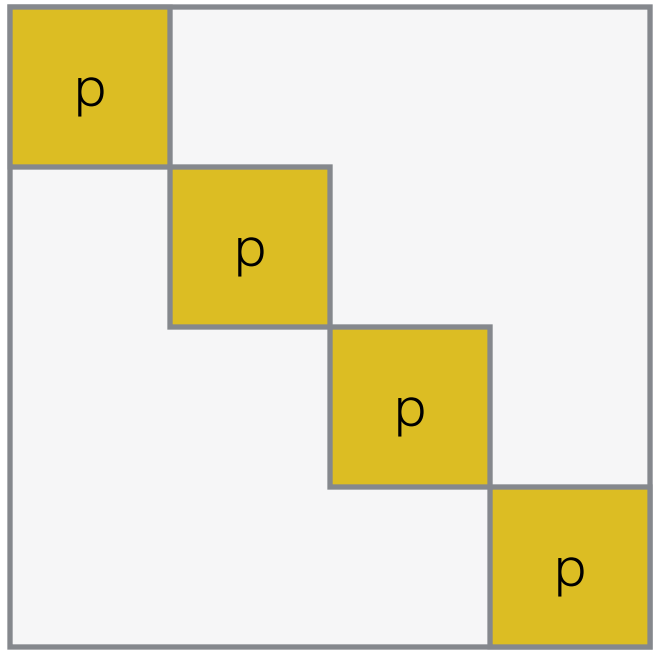
The zoo of possible structures is diverse and interesting!
Build intuition: what do these networks look like?



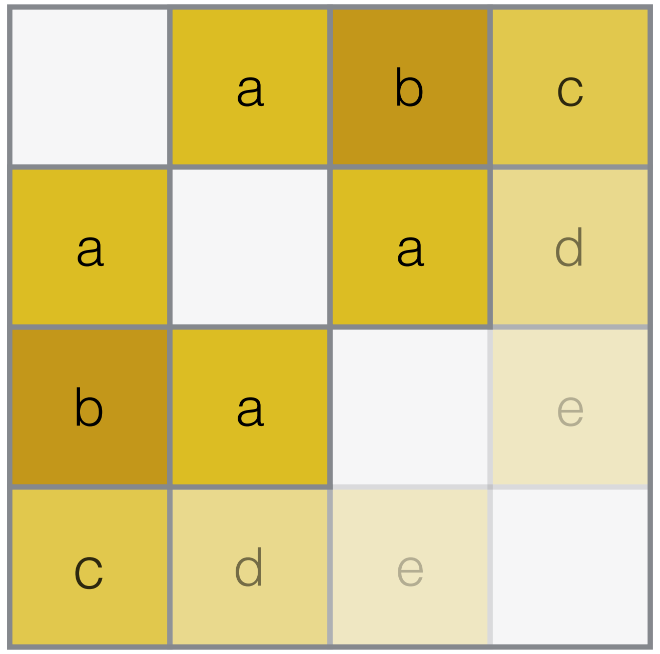
Assortative   Disassortative   Ordered   Core-periphery

Beyond assortativity: block models

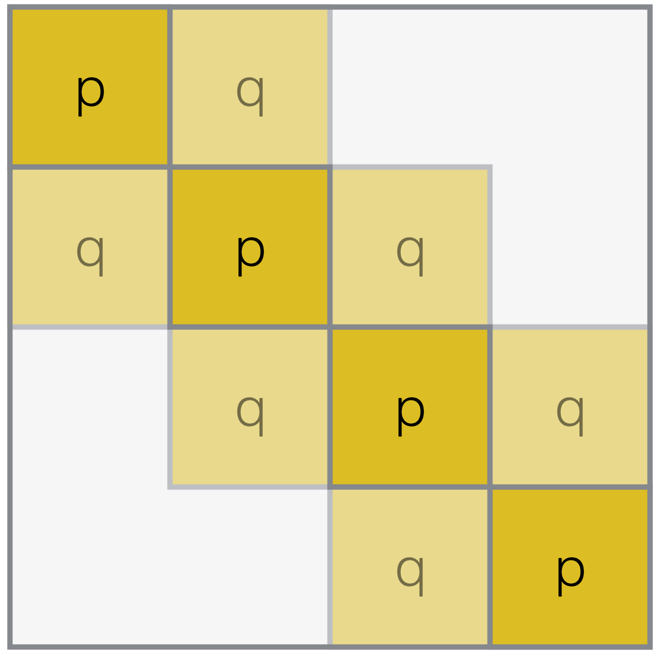
What do these have in common?



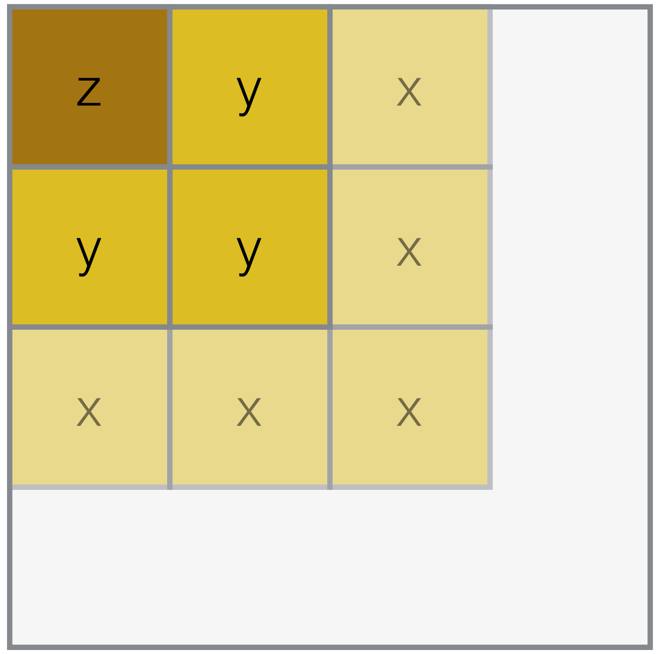
Assortative



Disassortative



Ordered



Core-periphery

Nodes are in groups with other nodes that *connect to other groups in similar ways.*

Key idea: all nodes in a group are *stochastically equivalent*.

Generative model approach

Generate the structure you wish to infer.

We like generative models because they open the door to inference:

$$\text{GM + parameters} \xrightarrow{\text{stochastic draw}} \text{Data}$$

$$\text{GM + parameters} \xleftarrow{\text{inference}} \text{Data}$$

In other words: let’s write down a recipe for generating block structure. 👨🏻‍🍳 👩🏻‍🍳

The stochastic block model

GM + parameters $\xrightarrow{\text{stochastic draw}}$ Data

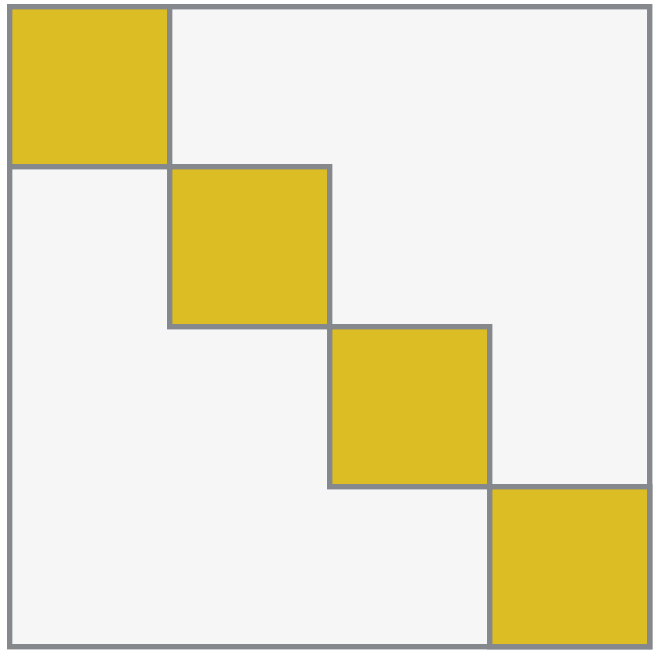
Assign each node to one of *B* blocks.

b_i

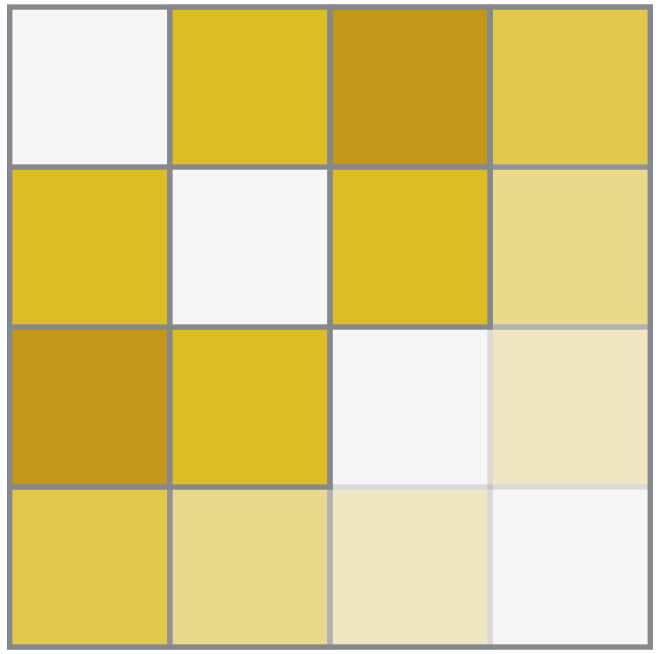
Let the probability that two nodes connect depend *only* on their blocks:

$$Pr(A_{ij}|b_i, b_j) = \omega_{b_i,b_j}$$

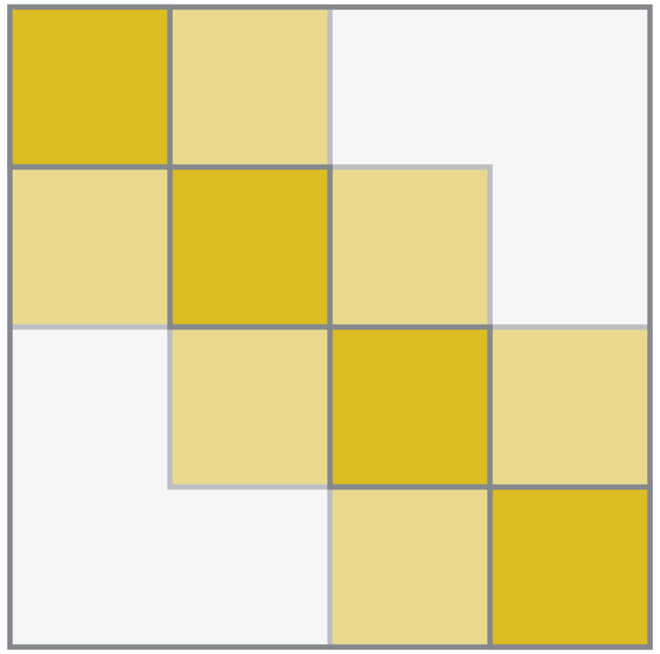
Then we can choose the matrix ω to have whatever structure we want!



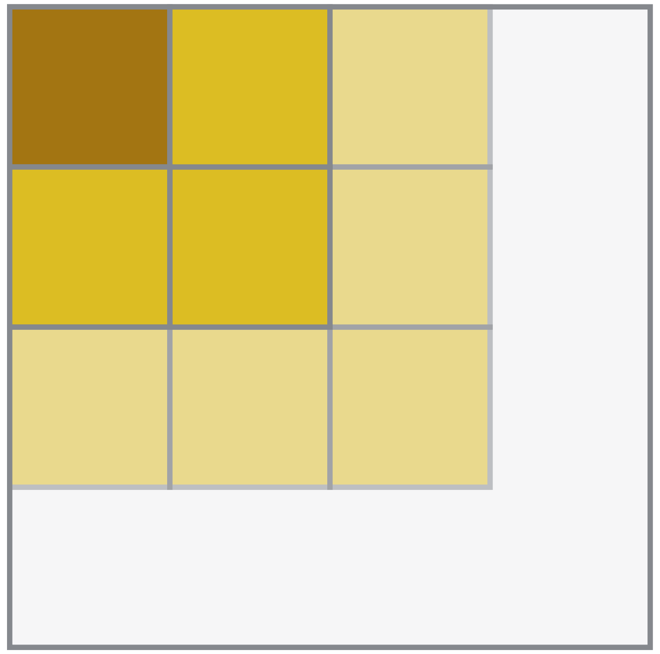
Assortative



Disassortative



Ordered



Core-periphery

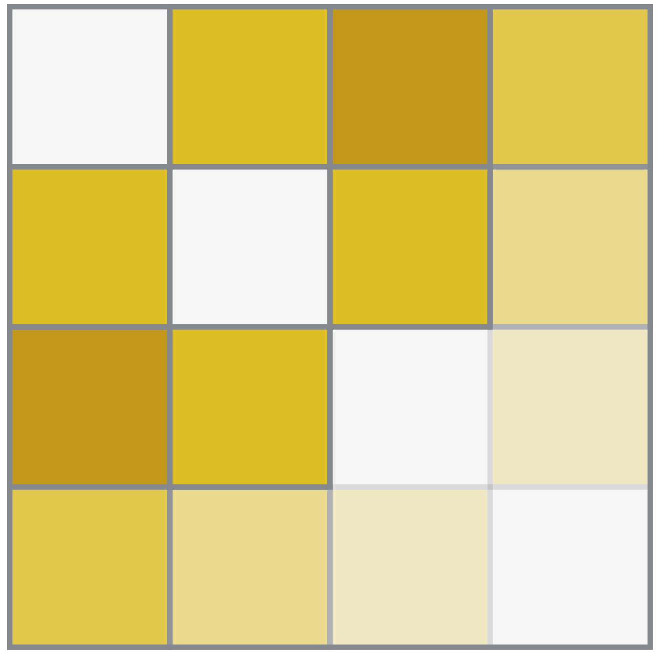
SBM inference

GM + parameters $\xleftarrow{\text{inference}}$ Data

no more math on slides 😭

(boringness prevention intervention)

Karrer, Newman. Stochastic blockmodels and community structure in networks. Phys. Rev. E 83, 016107 (2011).



example matrix of parameters, *B=4*

Summary:

1. Write down the SBM *likelihood function* for a fixed number of blocks *B*.
2. Maximize the likelihood with respect to matrix parameters.
3. Search over divisions into *B* blocks to find the best blocks.

The problem with parameterized models…

You have to choose their parameters!

How shall we choose *B*, the number of blocks?

Hint: we can't simply maximize the likelihood over all choices of *B*:
    Why? If we place each node in its own community, we can get Likelihood=1.
    [Actually, this wouldn't model the data at all: it would *memorize* it.]

We need a way to penalize the complexity of the model. Any ideas?

Description length & Occam's razor

The **Description Length** of a message is:
bits required to send the compressed message + # bits in encoding scheme.

Occam's razor: among all possible explanation for a phenomenon, choose the simplest one. Therefore, choose the model with Minimum Description Length (MDL).

The stochastic block model also has a Description Length:

$$\boxed{\Sigma} = \boxed{\mathcal{S}} + \boxed{\mathcal{L}}$$

description length = entropy of data, given the model (fit SBM) + entropy of model

Consider the original problem:
what happens to this equation when I increase the number of blocks B?

Peixoto. Entropy of stochastic blockmodel ensembles. Phys. Rev. E 85, 056122 (2012).
Peixoto. Parsimonious Module Inference in Large Networks. Phys. Rev. Lett. 110, 148701 (2013).

MDL criterion suggests an algorithm:

Fit the SBM with 1 block and record the Description Length.
Fit the SBM with 2 blocks and record the Description Length.
…
when the Description Length starts to increase, go back one step and stop.*

Bonus: what happens if I *try to trick you* and give you a *random network with no blocks*?

MDL approach will tell you: your network is a random network with *one* block.

*Actually, use something clever, like Golden Ratio / Fibonacci search
Press et al. Numerical Recipes: The Art of Scientific Computing, (Cambridge University Press, Cambridge, England, 2007), 3rd ed.

So how does the search part work?

Markov-chain Monte Carlo:

Wander from one partition to another partition by proposing to take a node from one group and move it to a new group.

> If this move increases the likelihood score, then keep the move.
> If this move decreases the likelihood score, then maybe keep it, depending on how bad it is.

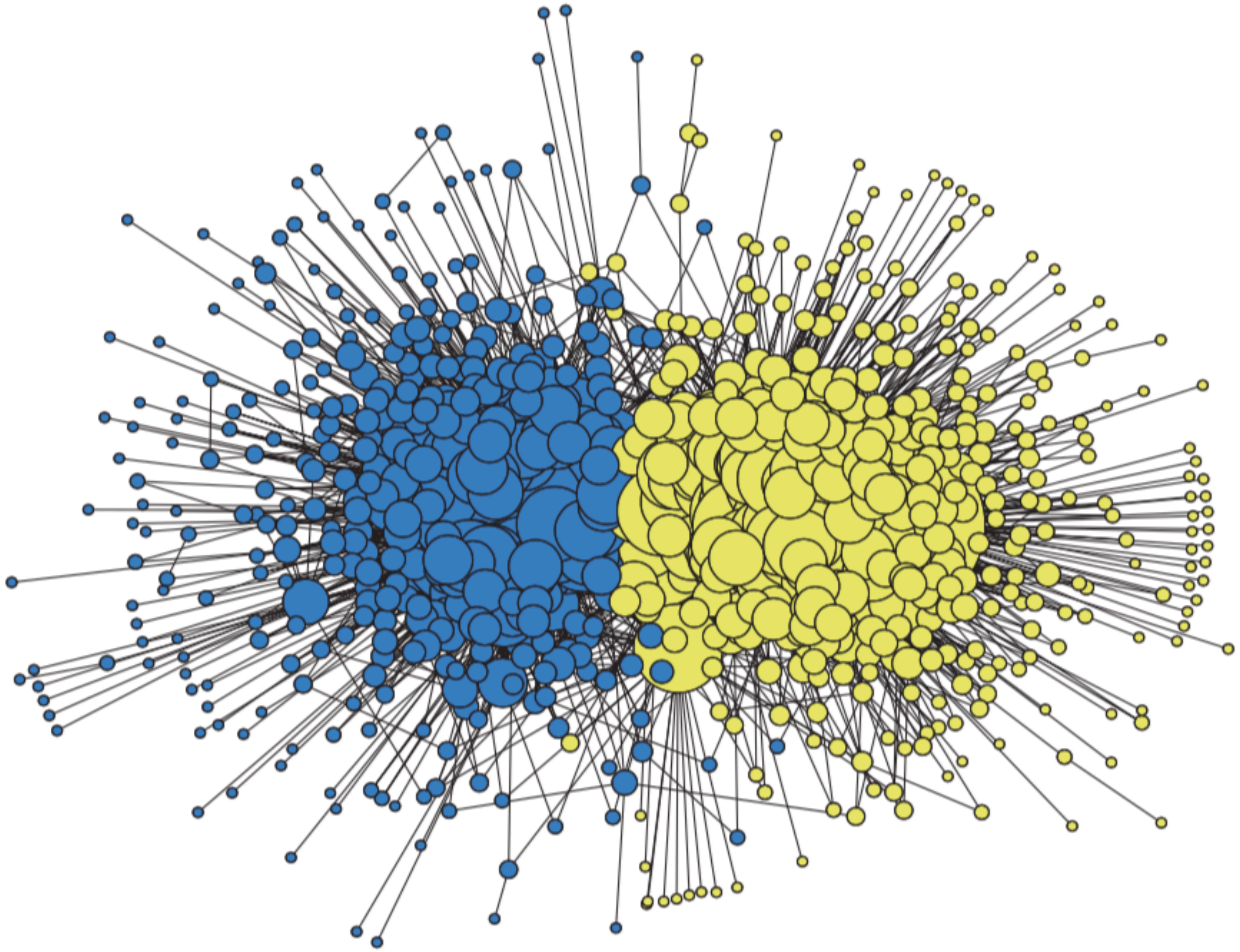
Thorough details in the documentation for graph-tool. https://graph-tool.skewed.de

Does it work?

Adamic & Glance mapped the link structure of USA political blogs in 2004.

Karrer & Newman used this network as a testbed for community detection using the SBM.

What does this say about the process that may be generating (or pruning?) the links in this network?



Division in political blog network. 👨🏽‍💻👩🏻‍💻

Karrer & Newman PRE 2011. https://arxiv.org/abs/1008.3926

Adamic & Glance KDD 2005. https://dl.acm.org/citation.cfm?id=1134277

Does it work?

In bipartite networks, we *know* the major split in the data already.

Methodologically, we found that exploiting this split improved speed and quality of the partitions we found.

Scientifically, this opened new directions to analyze (and understand) evolutionary constraints on malaria parasites.

Genes & substrings,
malaria immune evasion

Larremore, Clauset, Buckee, *PLoS Comp Biol*, 2013.
Larremore, Clauset, Jacobs, *Physical Review E*, 2014.

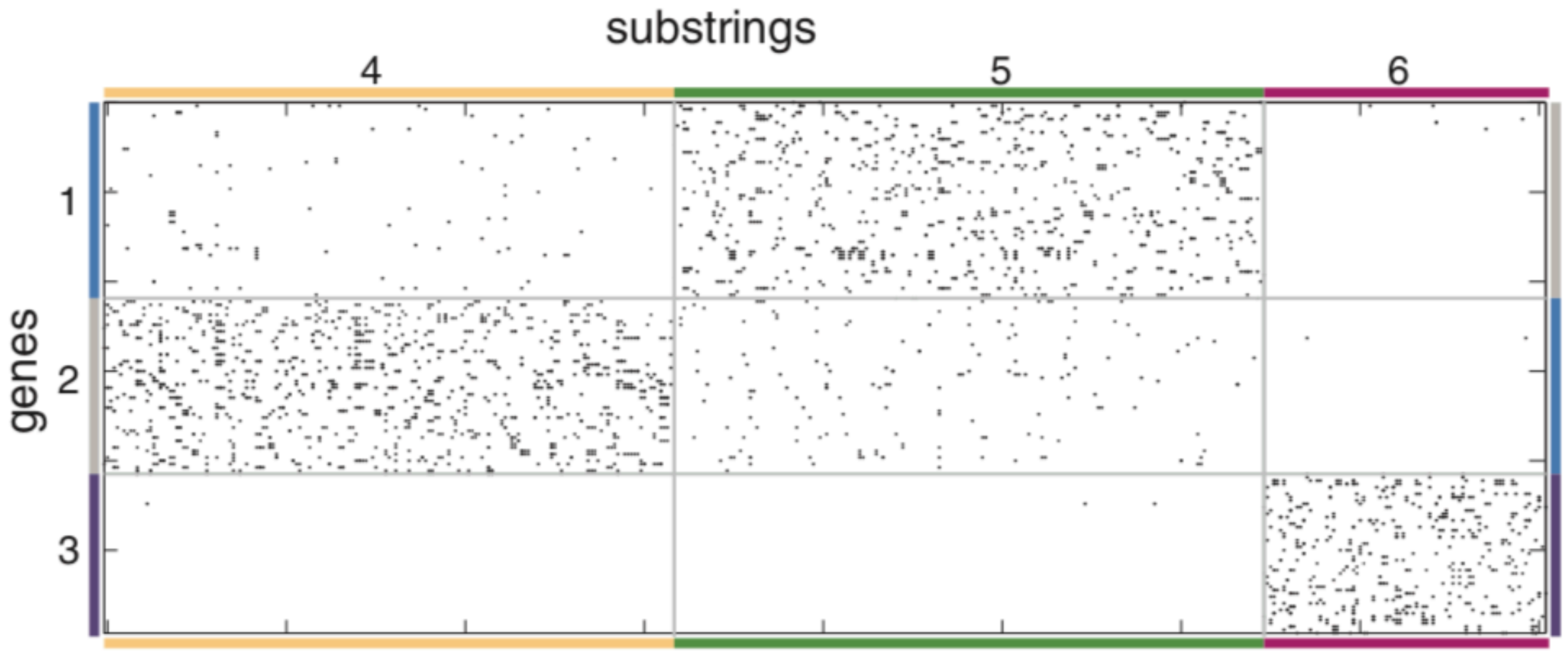
Code by Tzu-Chi Yen https://github.com/junipertcy/det_k_bisbm.
Figure http://danlarremore.com/webweb/

Does it work?

In bipartite networks, we *know* the major split in the data already.

Methodologically, we found that exploiting this split improved speed and quality of the partitions we found.

Scientifically, this opened new directions to analyze (and understand) evolutionary constraints on malaria parasites.



Genes & substrings,
malaria immune evasion

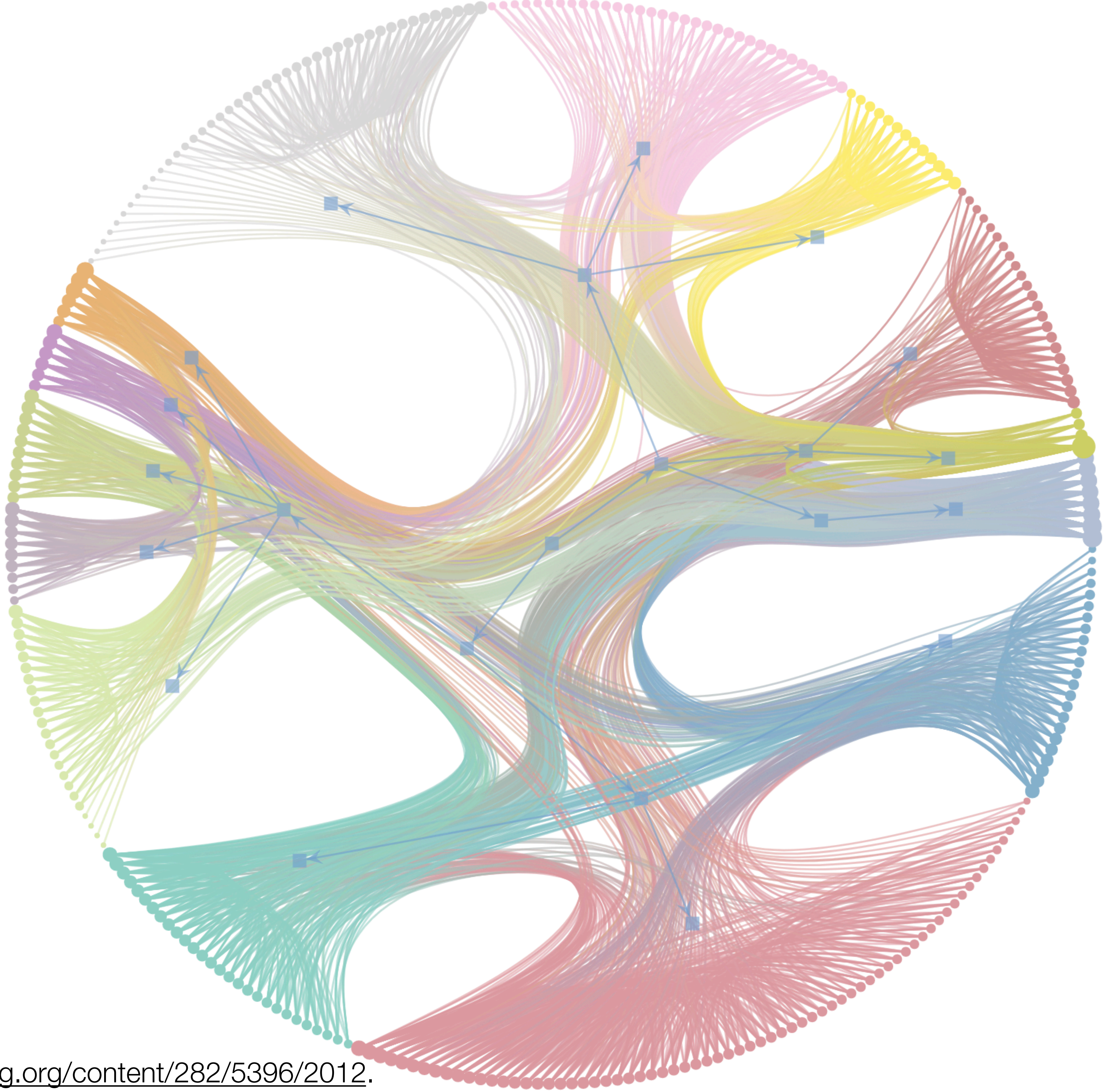
Larremore, Clauset, Buckee, *PLoS Comp Biol*, 2013.
Larremore, Clauset, Jacobs, *Physical Review E*, 2014.

Does it work?

C. elegans neuronal network.

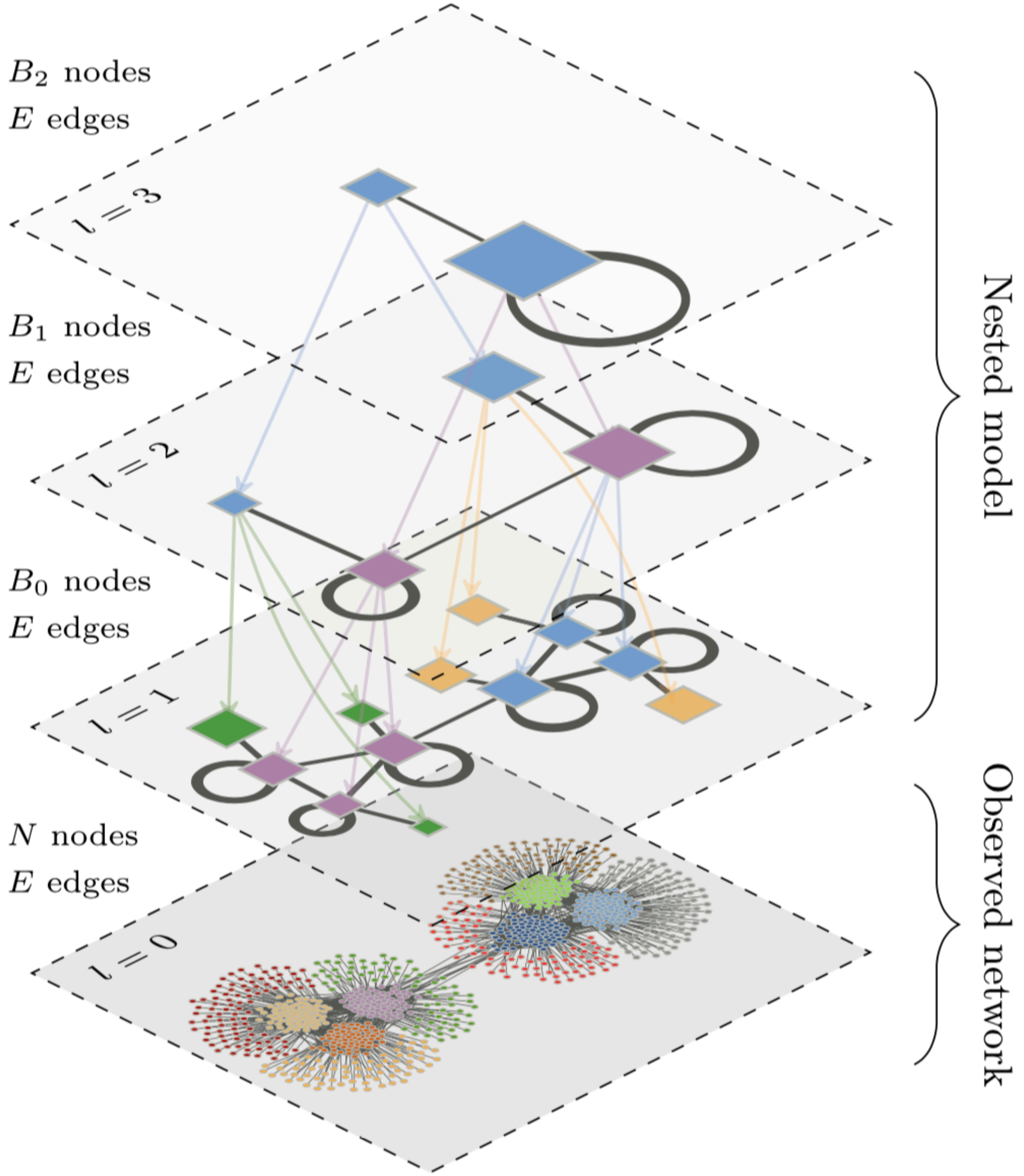
297 neurons, completely mapped. The neurons do not fire action potentials, and do not express any voltage-gated ion channels.

Note the different layout…



C. elegans 1st *multicellular* genome.1998. http://science.sciencemag.org/content/282/5396/2012.
Bacteriophage 1st genome: https://www.nature.com/articles/265687a0

Advanced topic 1: hierarchical communities



Don't minimize the description length of the data and the model…

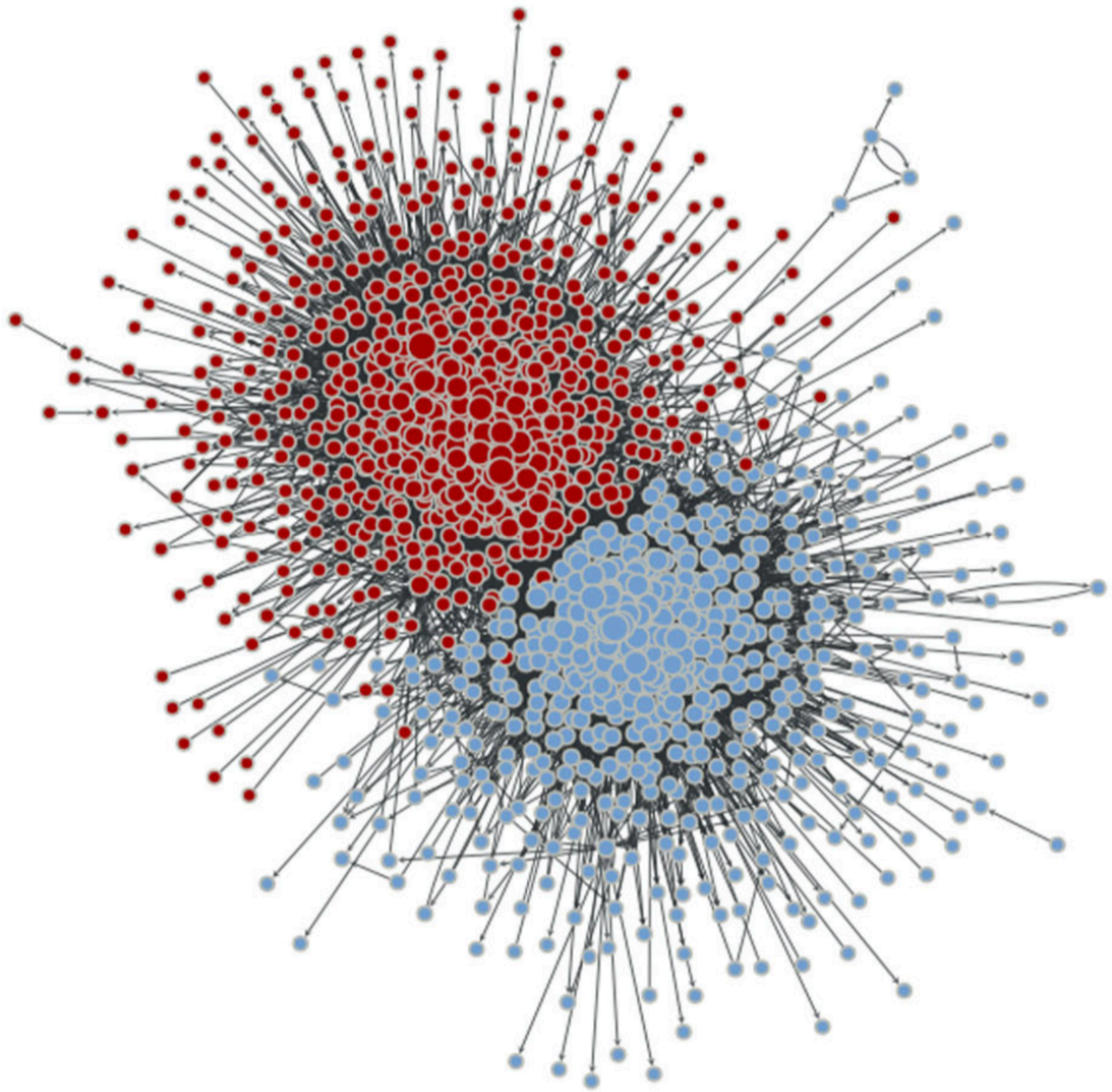
Model the model as well. Why?

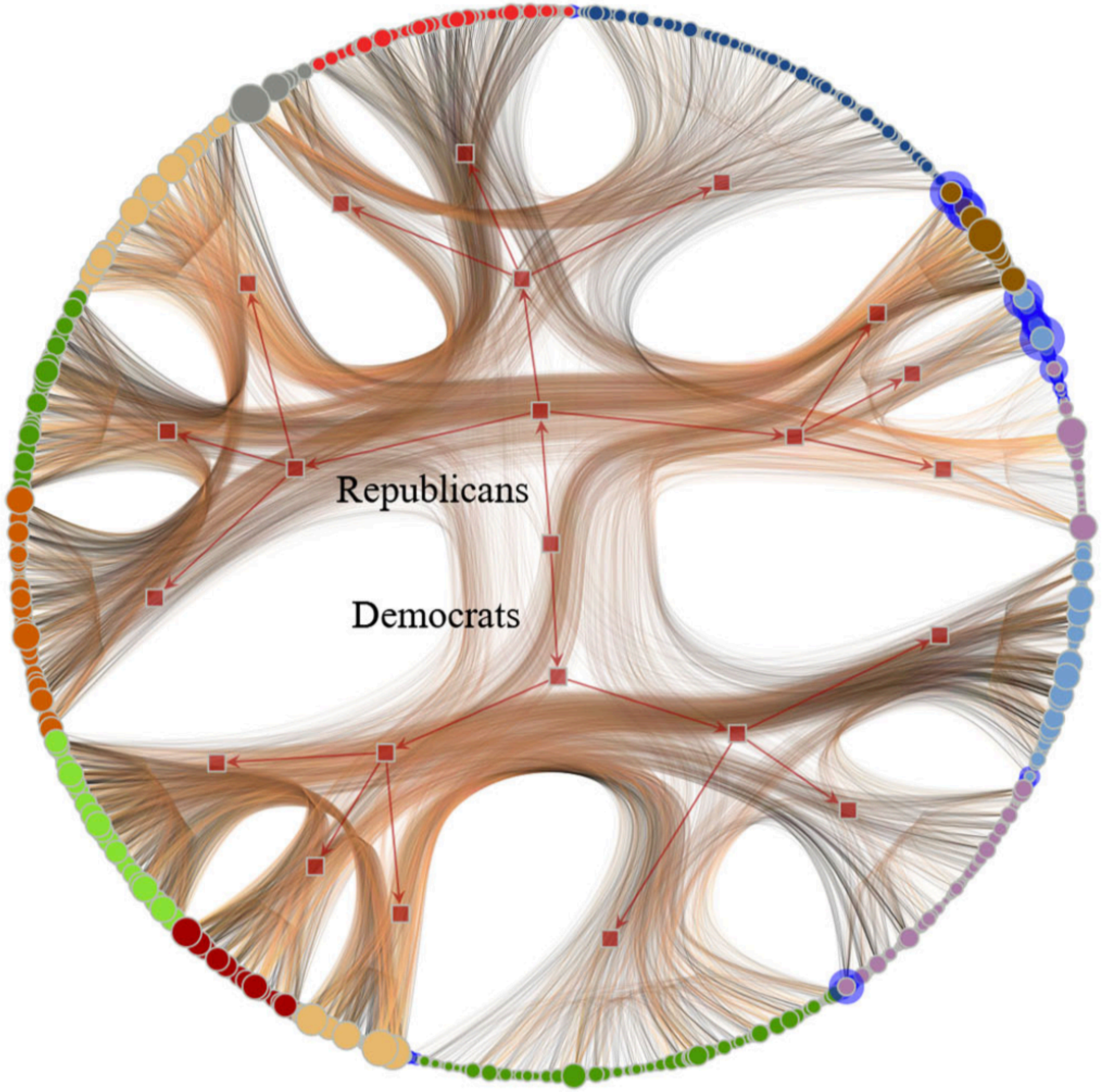
If we compress the model, we can afford a bigger model, but a lower overall cost.

Except now, the description length includes two models. Or three? Or?

Peixoto. Hierarchical Block Structures and High-Resolution Model Selection in Large Networks. Phys. Rev. X 4, 011047 (2014).

Advanced topic 1: hierarchical communities



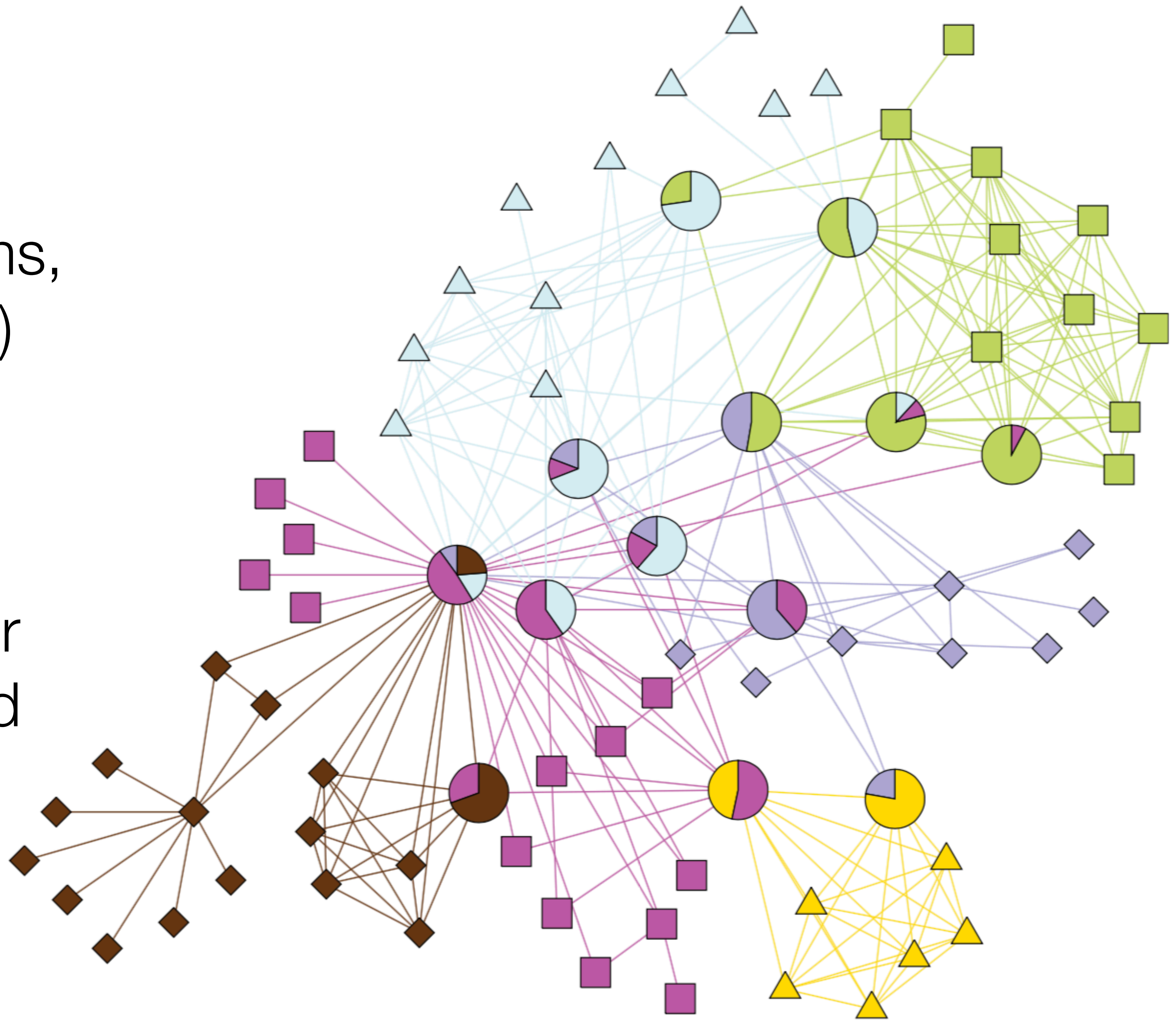


Peixoto. Hierarchical Block Structures and High-Resolution Model Selection in Large Networks. Phys. Rev. X 4, 011047 (2014).

Advanced topic 2: mixed-membership

Nodes are often pulled between communities. (Or in real social systems, individuals belong to multiple groups.)

“Mixed membership” models allow for that, by assigning *links* to groups, and assigning nodes to groups based on their links.



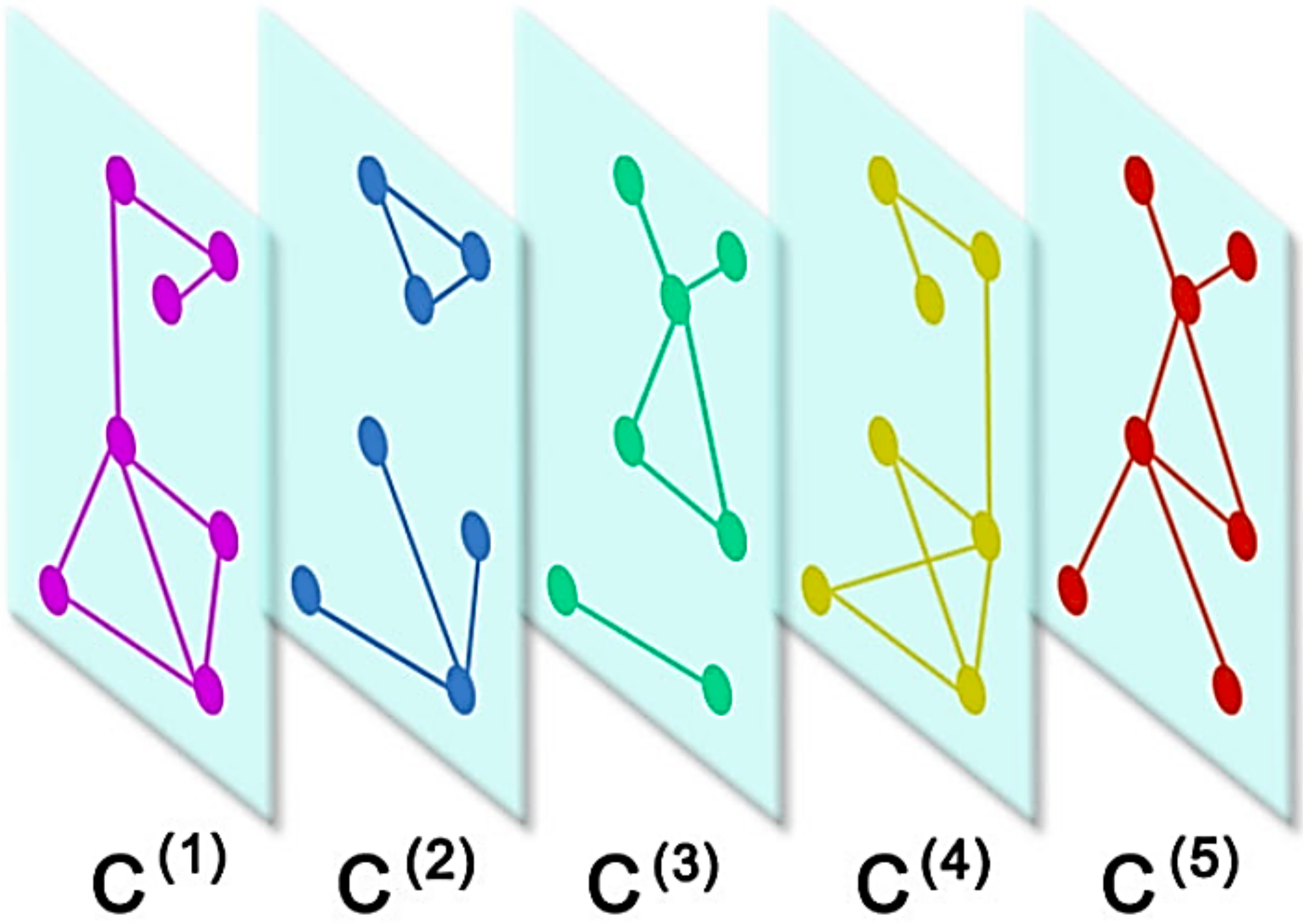
Ball, B., Karrer, B. & Newman, M. E. J. Efficient and principled method for detecting communities in networks. Phys. Rev. E 84, 036103 (2011).

Advanced Topic 3: multilayer networks

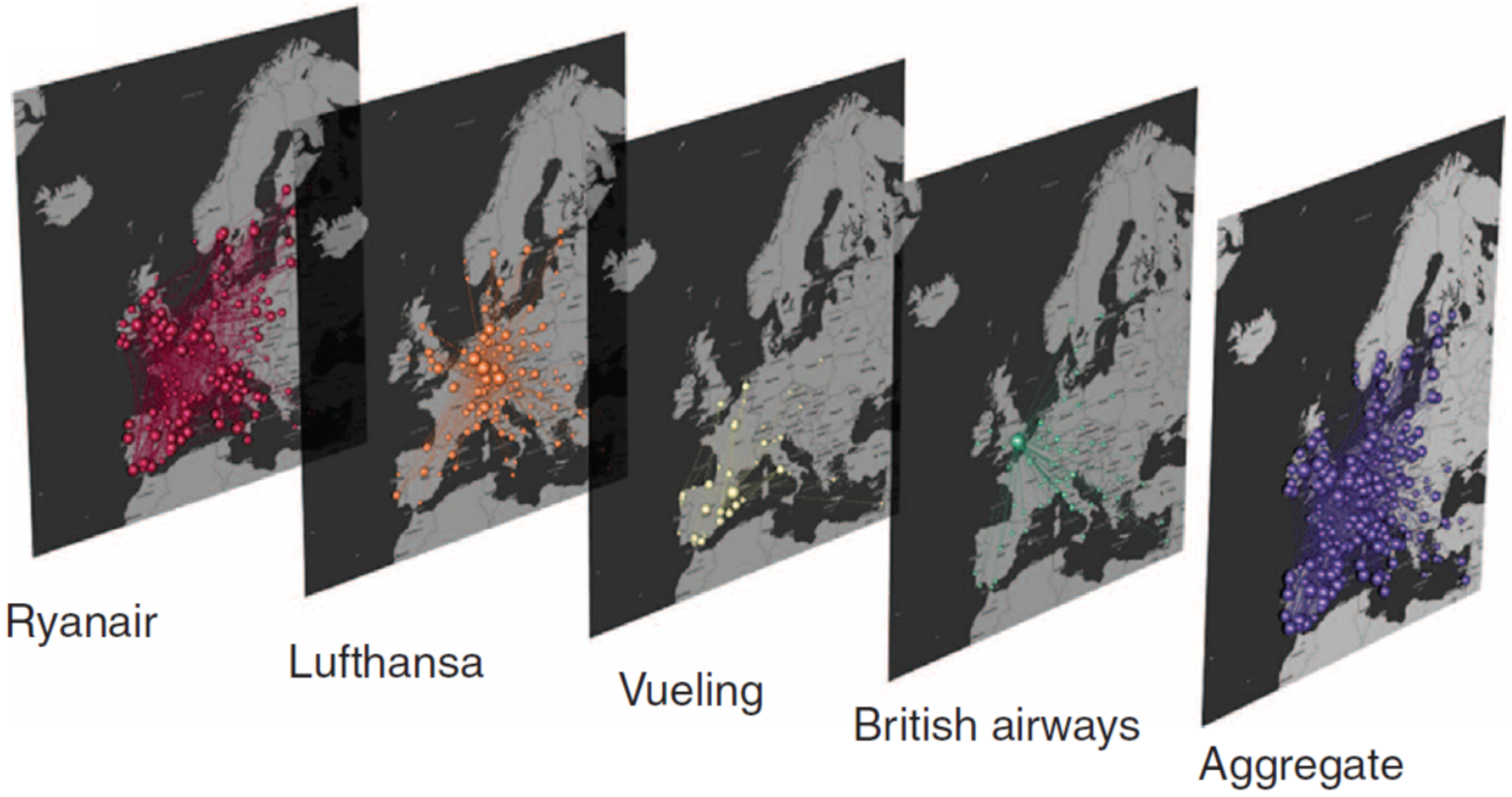
In single-layer networks:
  nodes and edges

In multilayer networks:
  nodes, edges, and layers

edges: different types of relationships
layers: each layer contains all edges of one type
nodes: same nodes in each layer



Multilayer network: air travel



traditional: booking with airline

disrupted: booking with kayak, expedia, etc

Multilayer network: community structure?

three key approaches:

1. **Non-generative**: modularity maximization; vary inter-layer strength.

   Mucha et al *Science* 2010. http://science.sciencemag.org/content/328/5980/876

2. **Generative**: SBM for each layer, but jointly model layers whenever their structures are sufficiently similar.

   Peixoto, T. P. Phys. Rev. E 92, 042807–15 (2015).

3. **Generative**: SBM for each layer, and model all layers simultaneously with same community structure, but allow relationships between groups to vary.

   De Bacco Power Larremore Moore. Phys. Rev. E 95, 1981–10 (2017).

1 is preferred if nodes appear/disappear over time.
2,3 are preferred to solve the *layer interdependence problem*

Other things to know about 1: “The Louvain Method”

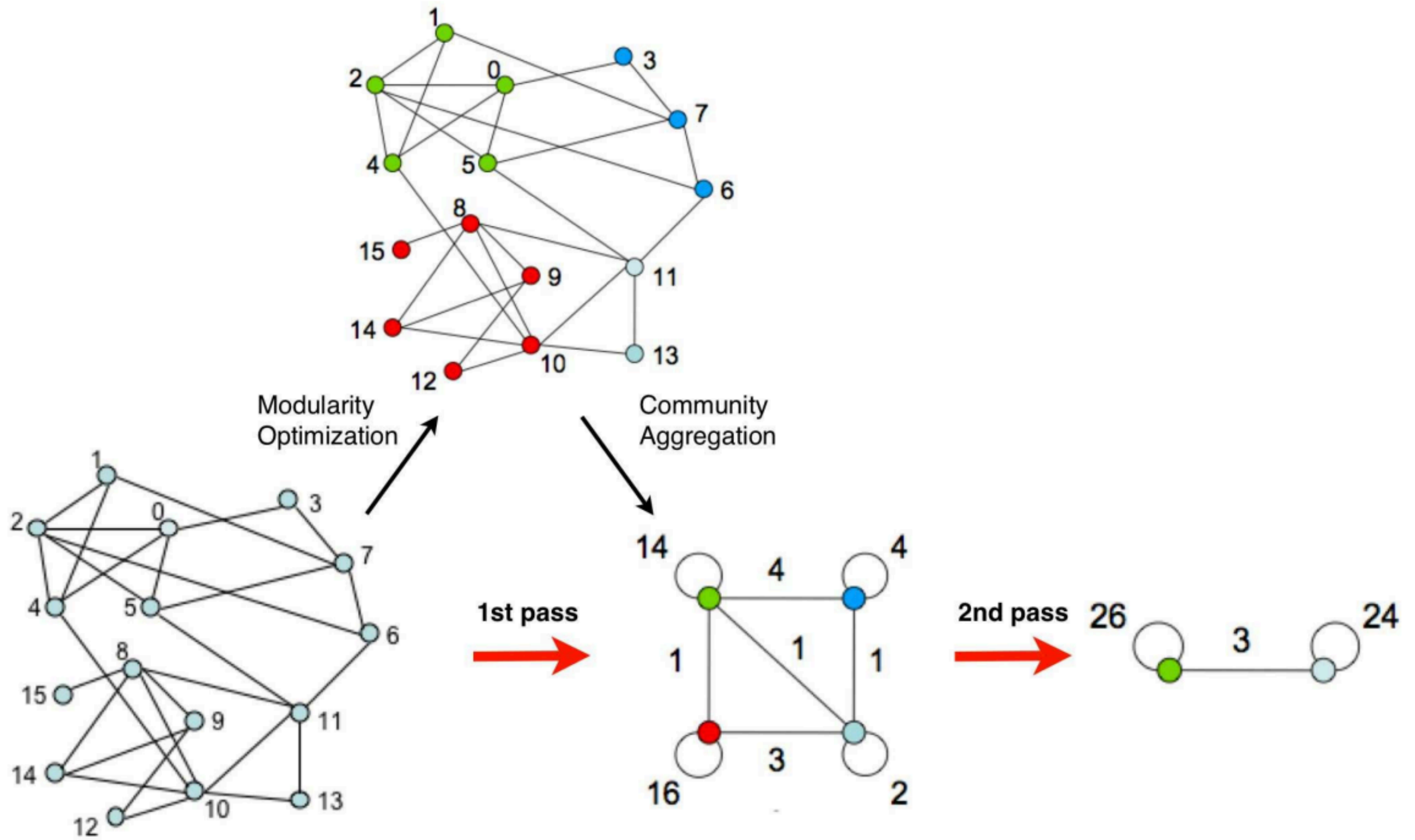
If your network is *really* big. (Millions of nodes, Billions of edges)

Take ClausetNewmanMoore’s approach for greedy Q maximization and find small groups.   Run the code again on those groups… And again…

Advantage: fast! big! 🚀

Disadvantage: inherits the assumptions of modularity. (clustering vs modeling)

6K citations. People like it!



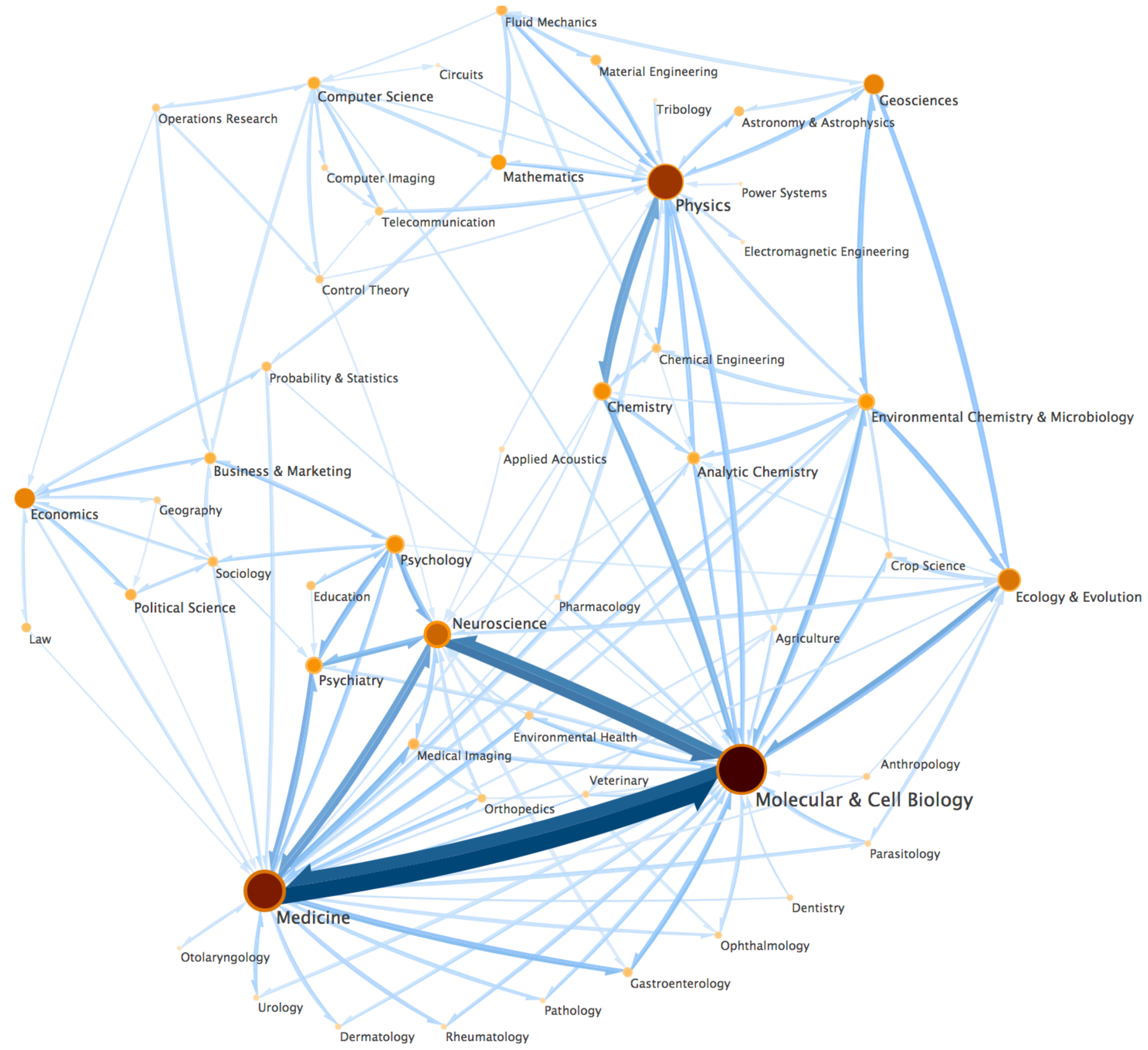
https://en.wikipedia.org/wiki/Louvain_Modularity   https://arxiv.org/pdf/0803.0476.pdf

Other things to know about 2: InfoMap

Imagine a random walker on a network.

A description of her walk can be compressed if the network has regions in which the random walker tends to stay for a long time.

Minimizing the “map equation” over all possible network partitions is the same as finding the best codebook.



http://www.mapequation.org/apps/MapDemo.html 🤩

http://www.mapequation.org/code.html

Outlook for community detection

Simply put, we have amazingly powerful tools that did not exist 15 years ago.

Many are principled, statistically rigorous, and we learn more all the time. Those that aren't statistically rigorous are really, really fast.

There is no multiple regression for networks.

"Controlling for C, how important is X in predicting Y?"

Tradeoffs between general and bespoke methods are still being explored.

Outside of SBM, Modularity, Louvain, Infomap, it's a wild west.

Methodologists are keen to be challenged by new problem types.

New scientific questions inspire new methods.

Rankings and linear hierarchies

Many uses for the same techniques. cf regression

Treat the network like a system:

Extrapolation. Make predictions for as-yet unseen nodes (in “space” or time).
Interpolation. Identify missing links.
Generalization. Nodes of this type are like others of the same type.

Treat the network like an artifact:

Mechanisms. How did this network arise? What rules governed its assembly?
Explanations. Coarse-graining or compression.

Treat the network like a means to an end; an intermediate data structure:

Useful division. Need ranking so that we can assign experimental treatments.
Simplification. Downstream regression model needs ranks or groups.

The idea of rankings—pervasive!

Assumptions:

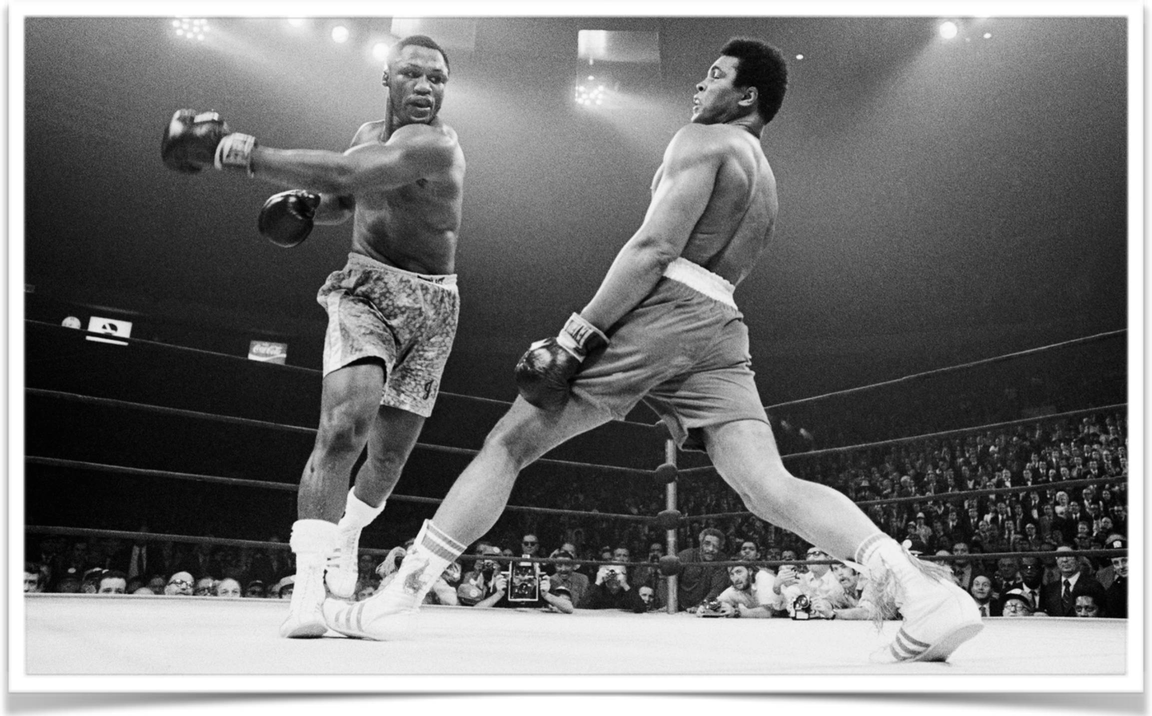
1. Competitors have some intrinsic quality (or vector of qualities).
2. Interactions can (stochastically) reveal differences in qualities.
3. Competitions are pair-wise. (Lee Sedol vs. AlphaGo; Astros vs. Dodgers)

In other words: outcomes are generated by a stochastic process, which is some function of the positions of the competitors.

Systems of dominance

social

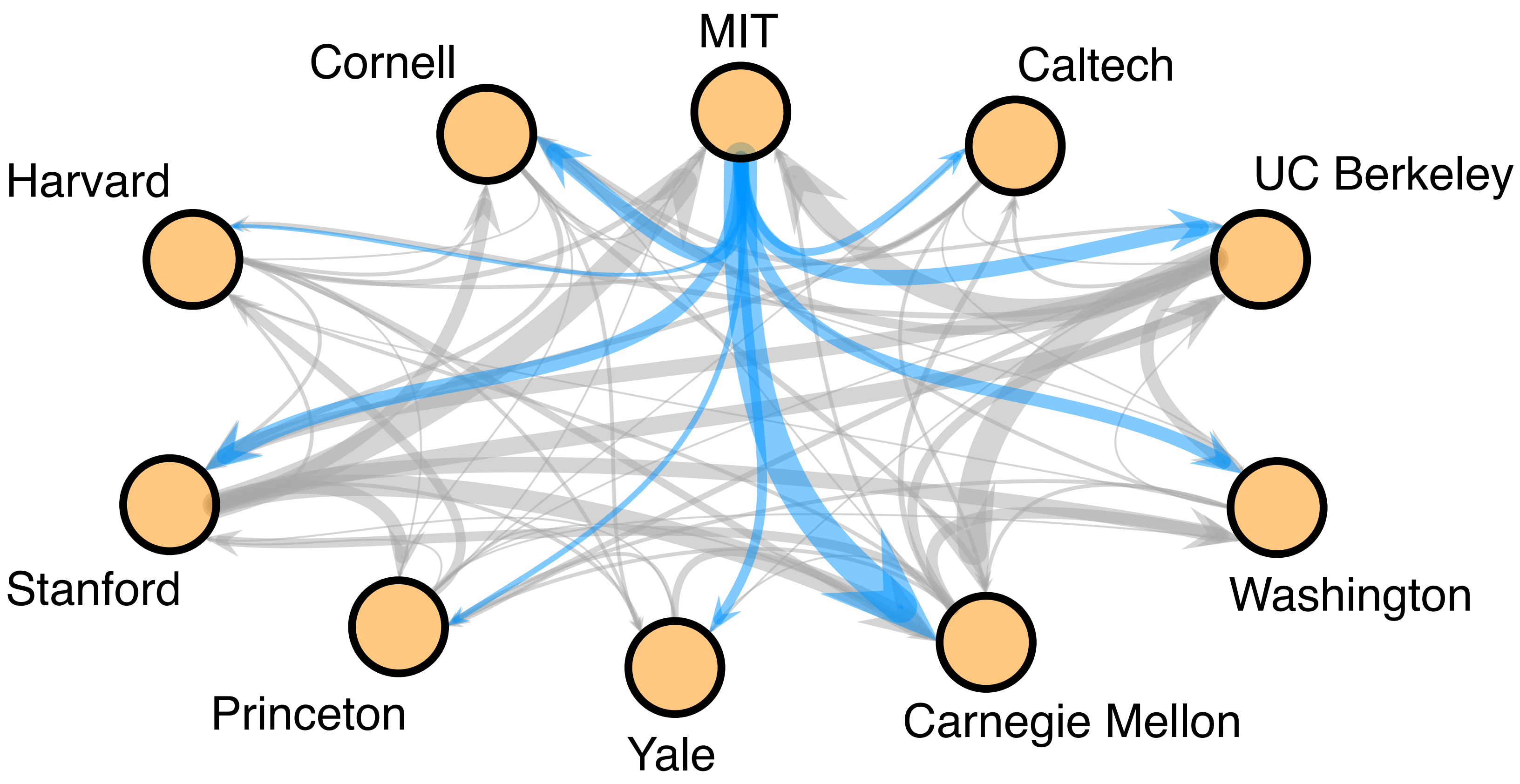
physical

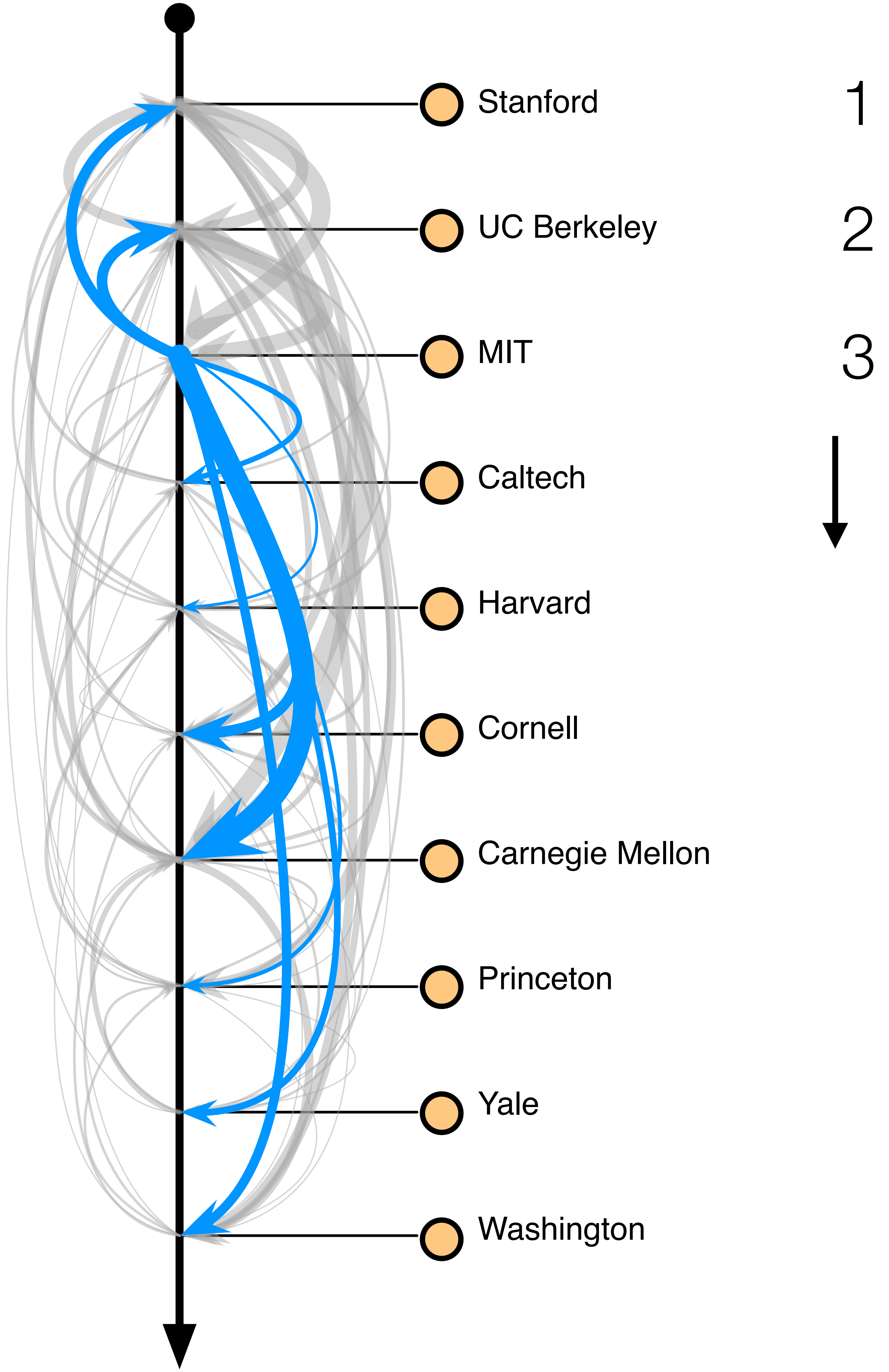


mental

financial

Systems of endorsement



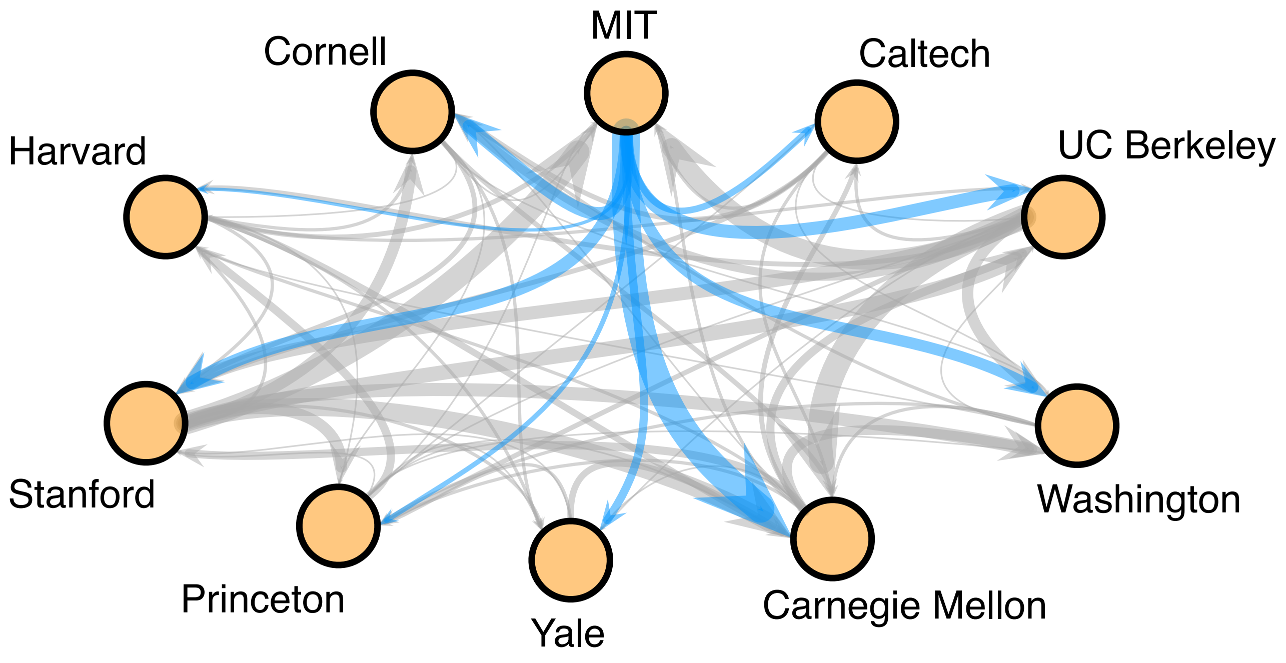


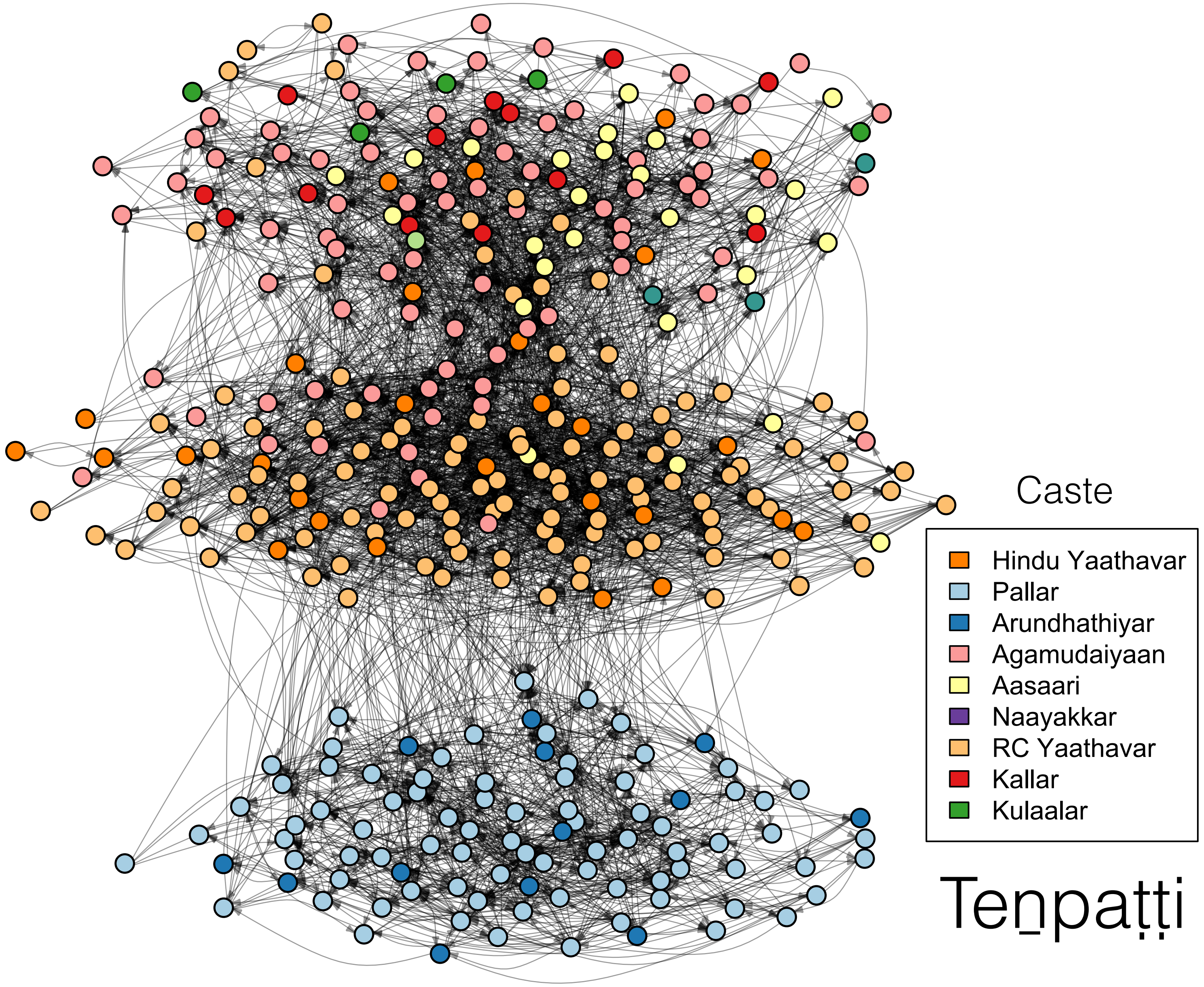
Assumptions:

1. Endorsers have some intrinsic quality.
2. Interactions can reveal differences in qualities.
3. Endorsements are pair-wise.

Clauset, Arbesman, Larremore. *Science Advances* 1, e1400005 (2015).

Systems of endorsement





Latent position can be revealed by **dominance** or **endorsement** interactions.

Figure: Larremore, Hébert-Dufresne, Power. *Draft.* Data: Power. *Nature Human Behaviour* 1, 0057 (2017)

Win-Loss is not satisfactory: schedule matters

Beating the grandmaster counts for more than beating a novice.

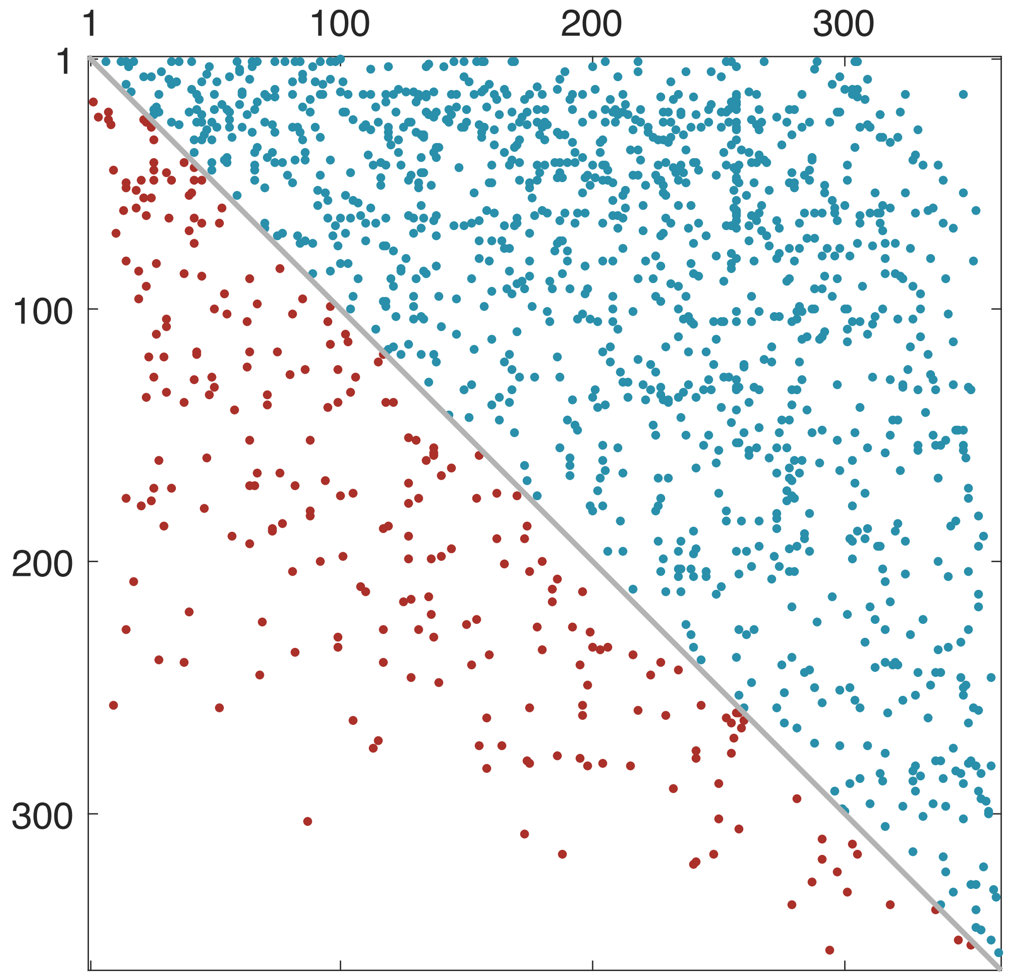
Win and loss tallies don't take this "schedule difficulty" into account. Put differently, win-loss records **leave information on the table.**

One way to make use of this information:

i* beats *j implies $s_i > s_j$

Therefore if we have a whole list of outcomes, we can try to find a total ordering that breaks as few of these implications as possible.

$A_{ij} =$ number of times that *i* beat *j*.



minimum violation ranking: sort *A*.

MVR: non-unique & rough optimization landscape

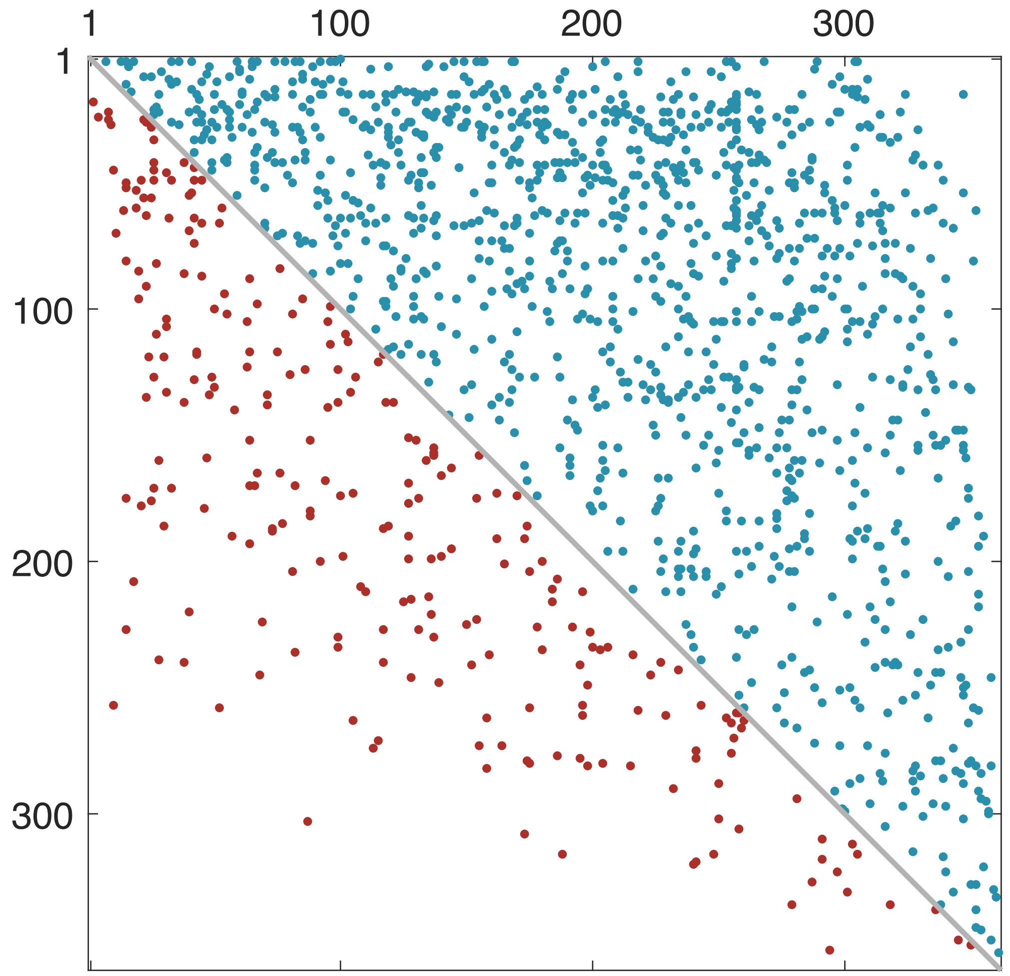
There is no guarantee of a unique minimizing ranking *s.*

Space of ordinal rankings has n! elements—usually use MCMC to search.

Slow.

Ordinal. No ties. No interpretability of rank differences.

What are other premises on which
we can base a ranking model?



minimum violation ranking: sort *A*.

Embeddings vs Orderings

Ordering place the nodes in order:

1, 2, 3, …

Embedding assigns a position to each node:

1, 1.2, 7, 20, 21, 21.2, …

Which one should I use?

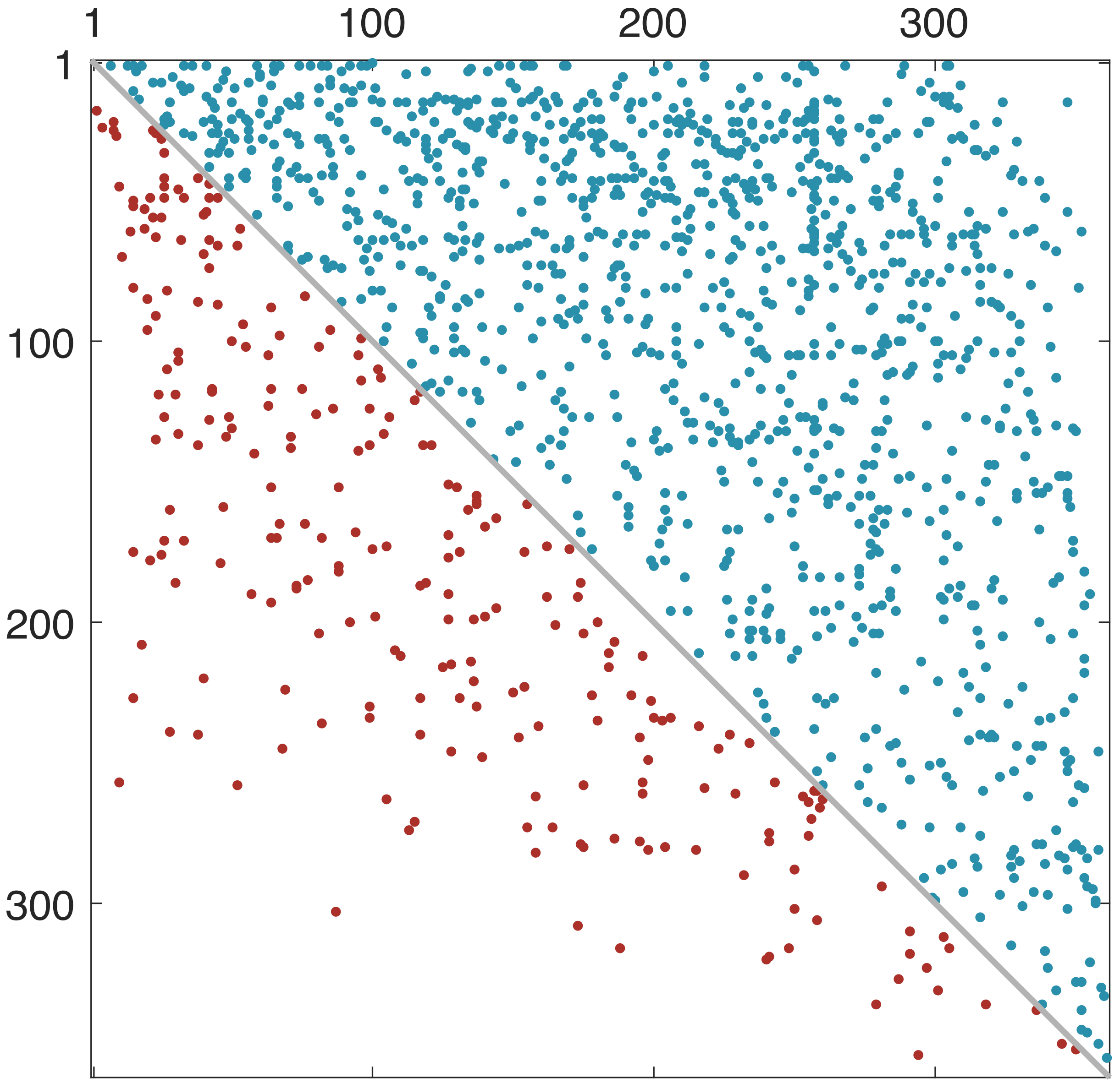
Depends on the use.

Is it possible for two nodes to occupy the same rank or position? If so, an embedding is more appropriate.

Consider that you can always go from an embedding to an ordering, if you have a rule for breaking ties.

Embeddings and Orderings: MVR & Agony

What if you allowed for ties and then ran Minimum Violation Ranking (MVR)? What would happen?



MVR: uniform cost (1 per edge).
Agony: generic cost function.

Embeddings and Orderings 1: PageRank

PageRank defines scalar rank recursively:

important pages are those that are linked to by important pages.

- Great at finding the top 3 but low interpretability of the PageRank scores.

The PageRank Citation Ranking: Bringing Order to the Web

January 29, 1998

Abstract

The importance of a Web page is an inherently subjective matter, which depends on the readers interests, knowledge and attitudes. But there is still much that can be said objectively about the relative importance of Web pages. This paper describes PageRank, a method for rating Web pages objectively and mechanically, effectively measuring the human interest and attention devoted to them.

We compare PageRank to an idealized random Web surfer. We show how to efficiently compute PageRank for large numbers of pages. And, we show how to apply PageRank to search and to user navigation.

The Anatomy of a Large-Scale Hypertextual Web Search Engine

Sergey Brin and Lawrence Page

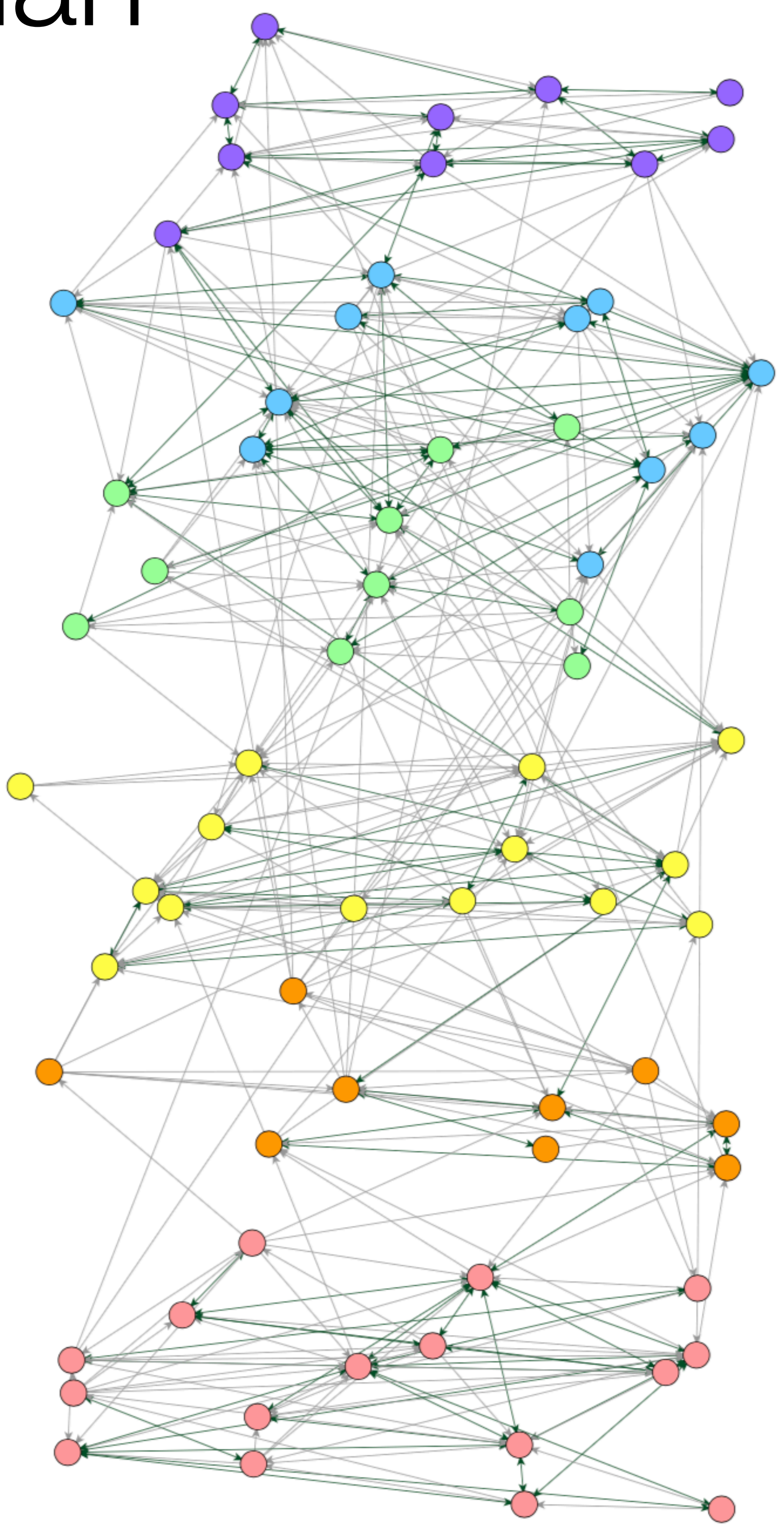
Computer Science Department,
Stanford University, Stanford, CA 94305, USA
sergey@cs.stanford.edu and page@cs.stanford.edu

Abstract

In this paper, we present Google, a prototype of a large-scale search engine which makes heavy use of the structure present in hypertext. Google is designed to crawl and index the Web efficiently and produce much more satisfying search results than existing systems. The prototype with a full

Eigenvector centrality on [A + very weak all-to-all links].

Embeddings and Orderings 2: Ball & Newman



Ball & Newman inferred ordinal rank via random graph + ranks model:

infer parameters of people's attachment preferences & ranks.

- Identified the need to learn from reciprocated friendships.
- Found that in AdHealth data, teens link to others of *nearby* social status.

Generative model + EM algorithm to fit it to data.

Integer ordering of nodes

Ball, Newman. Network Science 1, 16-30 (2013)

Embeddings and Orderings 3: Niche Models

Niche Models embed species in a latent space based on feeding preferences:

most species feed from narrow range in a 1-dim. space (~body size).

- Great for food webs. Inference models v slow for all but small networks.

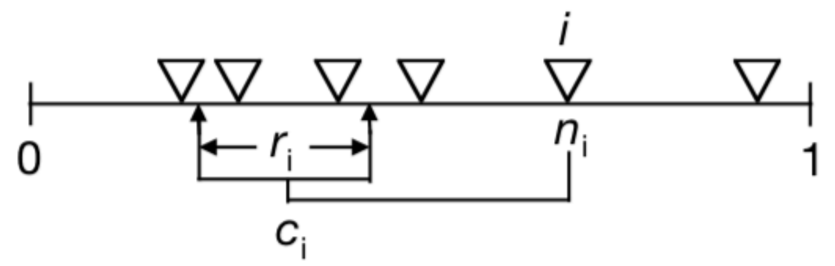


Figure 1 Diagram of the niche model. Each of ***S*** species (for example, ***S*** = 6, each shown as an inverted triangle) is assigned a 'niche value' parameter (n_i) drawn uniformly from the interval [0,1]. Species ***i*** consumes all species falling in a range (r_i) that is placed by uniformly drawing the centre of the range (c_i) from [r_i/2, n_i]. This permits looping and cannibalism by allowing up to half of r_i to include values $\geqslant n_i$. The size of r_i is assigned by using a beta function to randomly draw values from [0,1] whose expected value is 2***C*** and then multiplying that value by n_i [expected $E(n_i) = 0.5$] to obtain the desired ***C***. A beta distribution with $\alpha = 1$ has the form $f(x|1, \beta) = \beta(1-x)^{\beta-1}$, $0 < x < 1$, 0 otherwise, and $E(X) = 1/(1+\beta)$. In this case, $x = 1-(1-y)^{1/\beta}$ is a random variable from the beta distribution if y is a uniform random variable and β is chosen to obtain the desired expected value. We chose this form because of its simplicity and ease of calculation. The fundamental generality of species i is measured by r_i. The number of species falling within r_i measures realized generality. Occasionally, model-generated webs contain completely disconnected species or trophically identical species. Such species are eliminated and replaced until the web is free of such species. The species with the smallest n_i has $r_i = 0$ so that every web has at least one basal species.

Williams & Martinez. Nature 404.6774 (2000).

Embeddings and Orderings 4: BTL

Bradley-Terry-Luce embed players in a 1D space. Outcome direction is simply:

$$P(i \to j) = \frac{\gamma_i}{\gamma_i + \gamma_j}$$

- Provably avoids <u>non-transitive</u> properties
- Great when lots of data per interaction.

Pairwise ranking is really nice for ordering large sets of preferences too, and this model specifically models the probability that the preference will be for *i* over *j*.

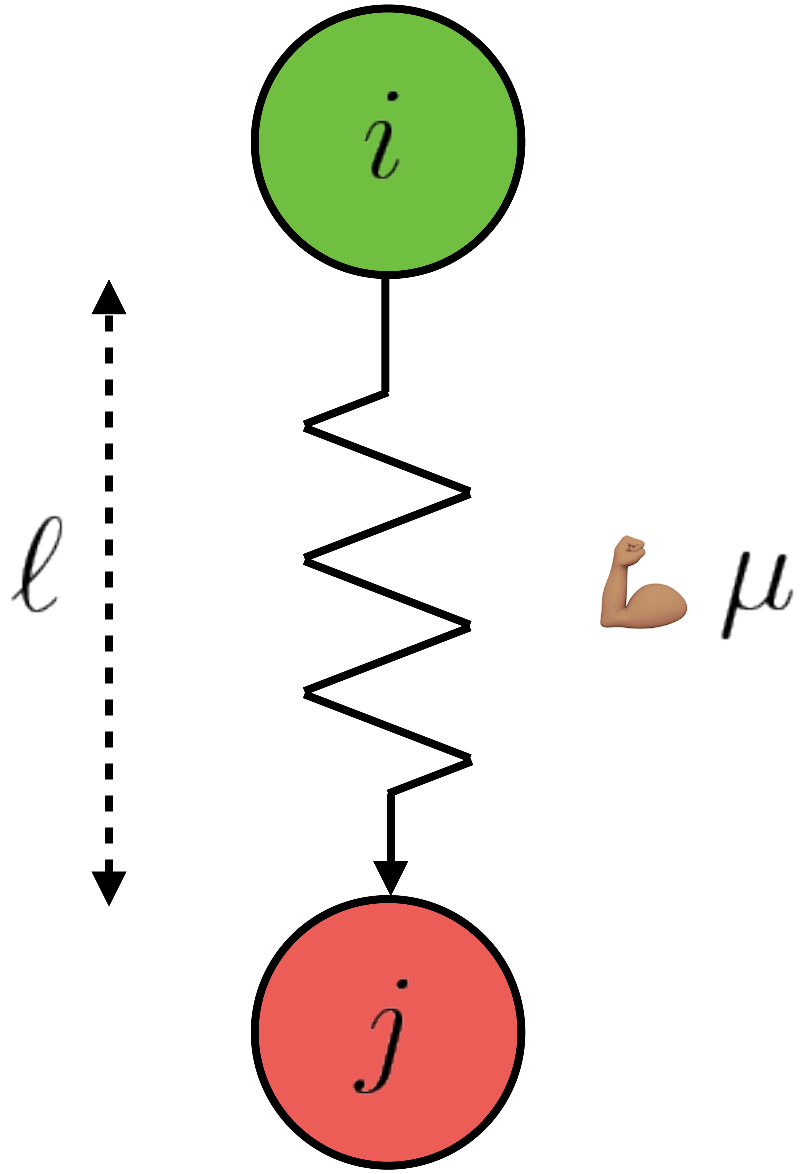
Iterative algorithms exist. Needs a little regularization so the winningest winners don't fly off to infinity.

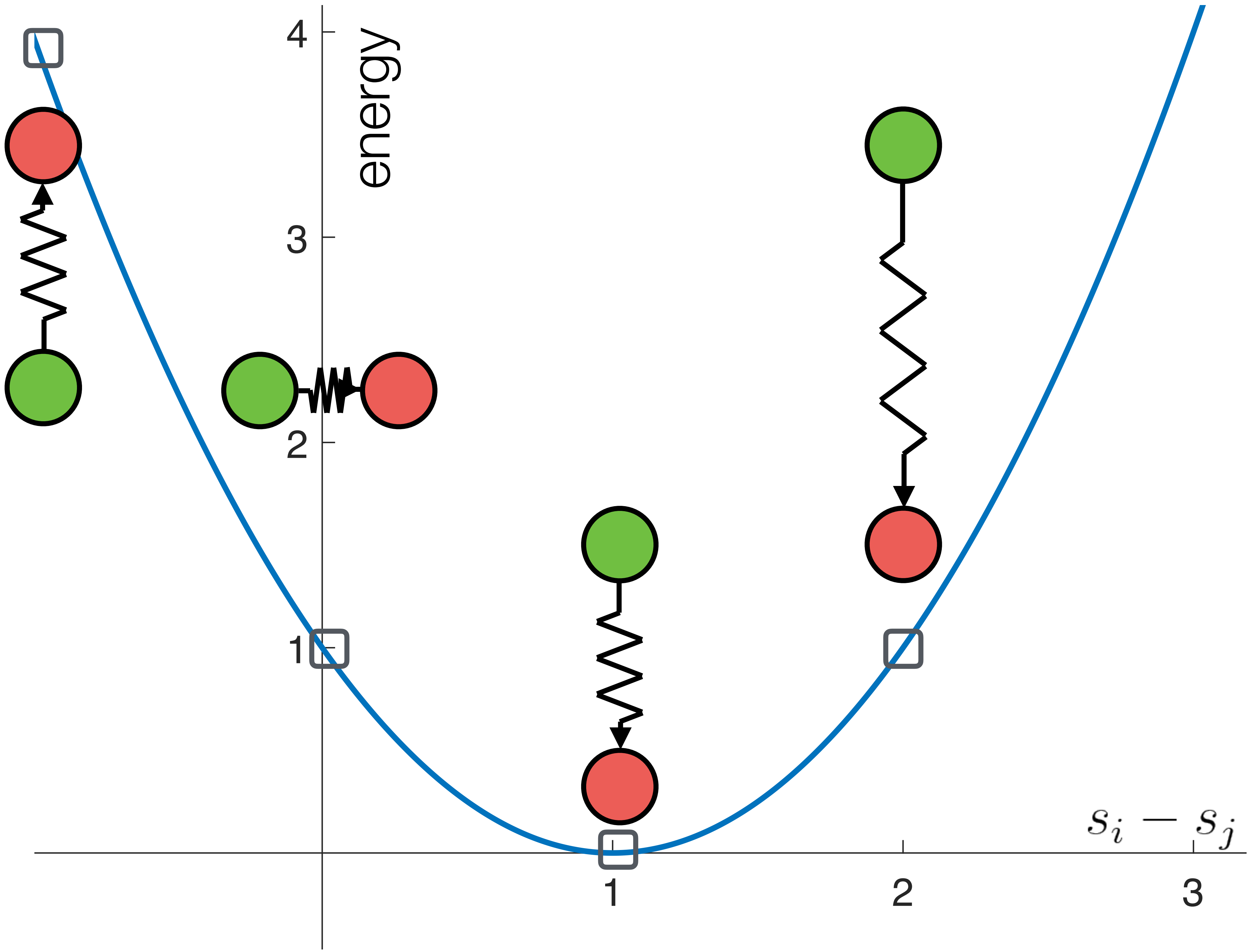
A few more minutes on:
SpringRank—fast, interpretable, predictive.

Honestly? Best to google "BTL model."

SpringRank

Each directed edge = directed spring





Relax and let the springs decide the ranks

$$H(s) = \frac{1}{2} \sum_{i,j=1}^{N} \mu_{ij} A_{ij} \left(s_i - s_j - \ell\right)^2$$

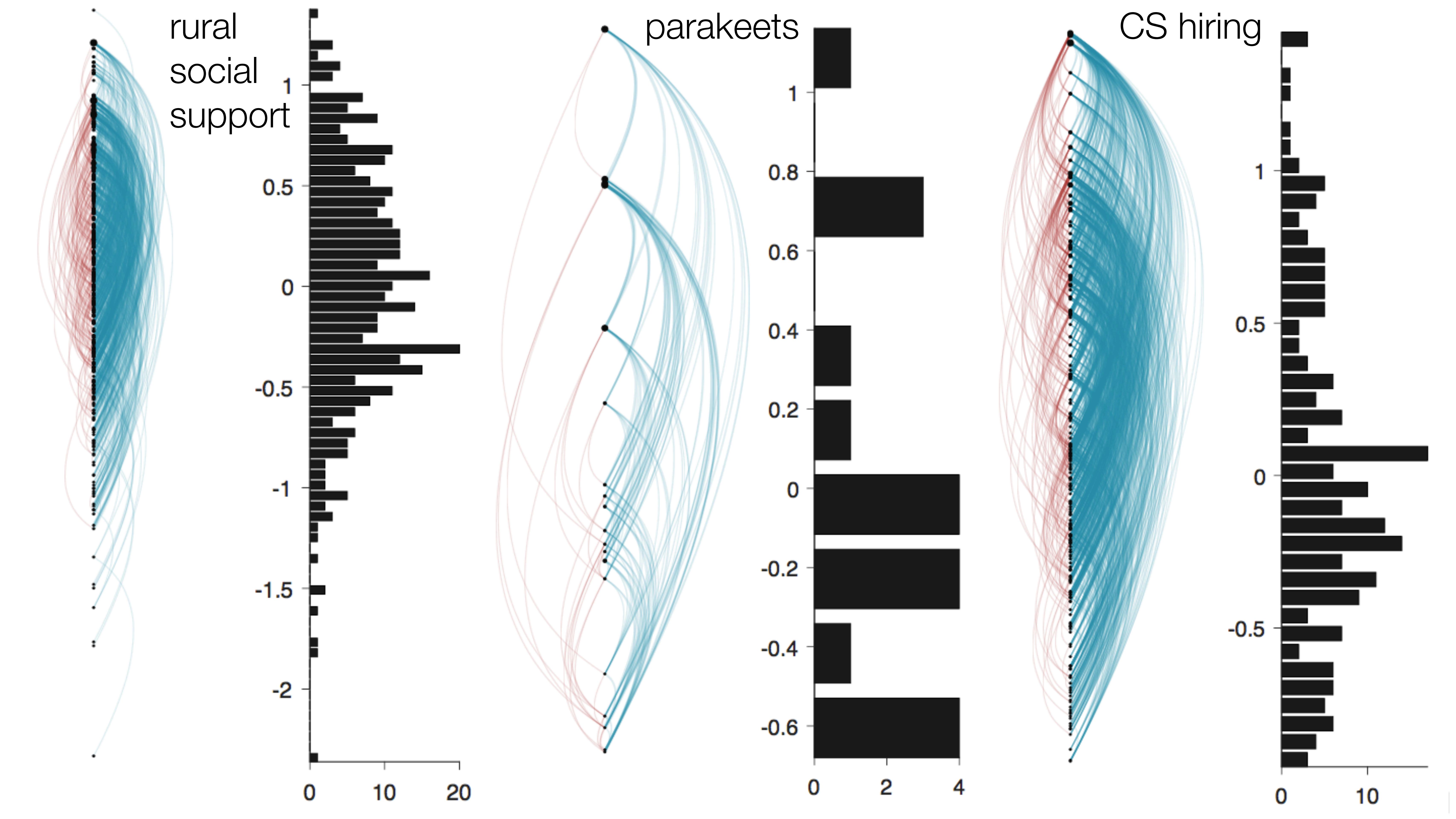
SpringRank Hamiltonian: energy of the system, given the positions *s.*

Because the springs are linear, the potential is quadratic.

The SR Hamiltonian is *convex* in s.

$$\nabla H(s) = 0$$

unique solution up to additive constant.


rural
social
support
1
0.5
0
-0.5
-1
-1.5
-2
0
10
20
parakeets
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
0
2
4
CS hiring
1
0.5
0
-0.5
0
10

Many uses for the same techniques. cf regression

Treat the network like a system:

Extrapolation. Make predictions for as-yet unseen nodes (in "space" or time).
Interpolation. Identify missing links.
Generalization. Nodes of this type are like others of the same type.

Treat the network like an artifact:

Mechanisms. How did this network arise? What rules governed its assembly?
Explanations. Coarse-graining or compression.

Treat the network like a means to an end; an intermediate data structure:

Useful division. Need groups so that we can assign treatments in an A/B test.
Simplification. Downstream regression model needs ranks or groups.

Goals for this talk:

1. **Why** do we look for large-scale structure? 🤔
2. **How** do we find communities and hierarchies? 🧐
3. **Where** can we read more details? 📚

Thank you

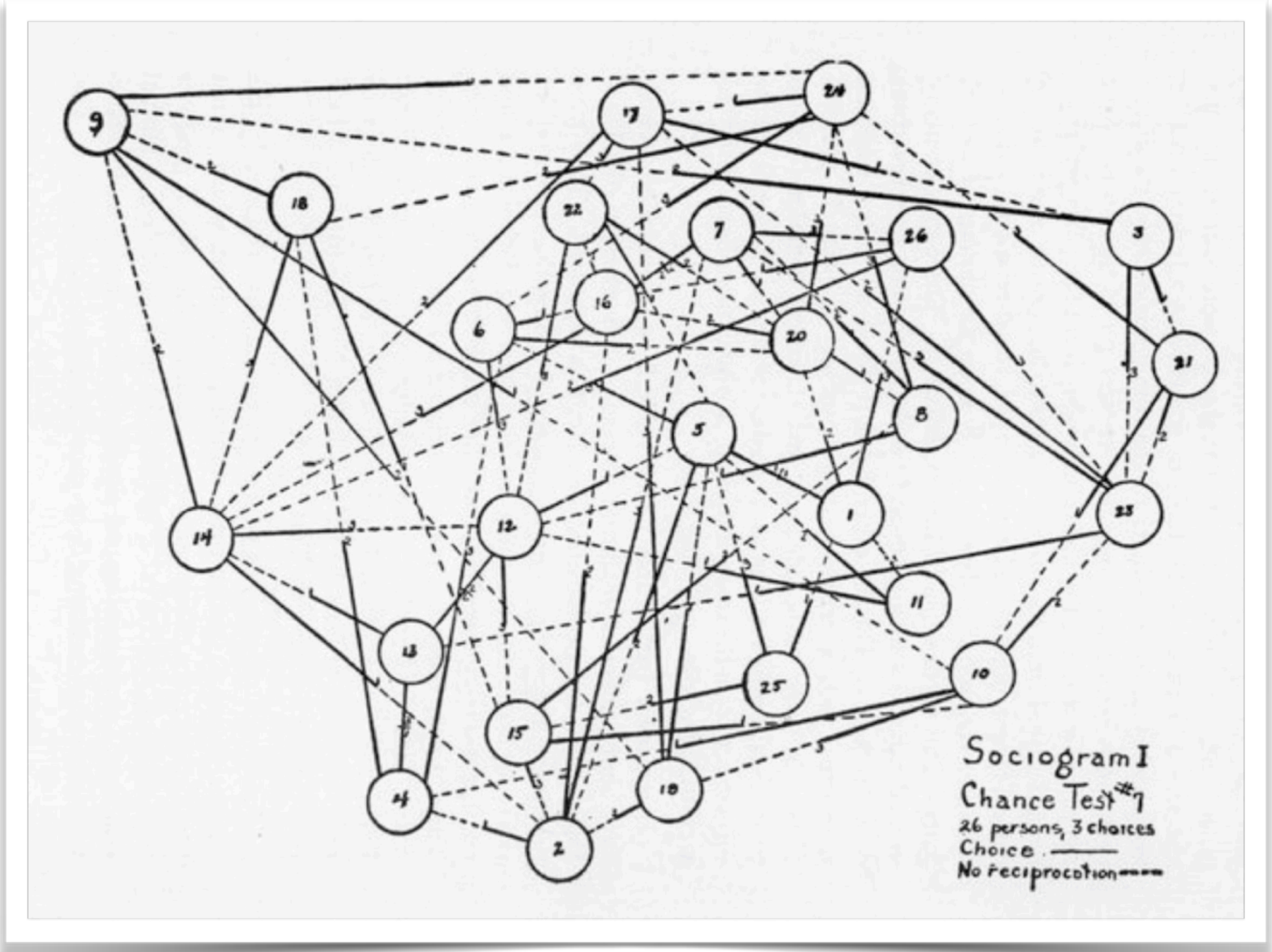
@danlarremore
daniel.larremore@colorado.edu

Aside: the birth of null models & chance sociograms

Who shall survive? Moreno, 1936

Moreno wondered if there were structural explanations for why certain young girls were running away from the school, and thought that sociographic analysis might hold an answer.

chance sociograms



SIAM Review: Configuring random graph models with fixed degree sequences. http://arxiv.org/abs/1608.00607

The Book: http://www.asgpp.org/docs/wss/Book%20VI/index.html

Johan Ugander’s Post: https://jugander.wordpress.com/2014/08/07/computational-perspectives-on-large-scale-social-networks-a-brief-history/

Aside:

Here is my favorite paper of all time:

JMLR: Workshop and Conference Proceedings 27:65–79, 2012 Workshop on Unsupervised and Transfer Learning

Clustering: Science or Art?

Ulrike von Luxburg ULRIKE.LUXBURG@TUEBINGEN.MPG.DE
Max Planck Institute for Intelligent Systems, Tübingen, Germany

Robert C. Williamson BOB.WILLIAMSON@ANU.EDU.AU
Australian National University and NICTA, Canberra ACT 0200, Australia

Isabelle Guyon ISABELLE@CLOPINET.COM
ClopiNet, 955 Creston Road, Berkeley, CA 94708, USA

http://proceedings.mlr.press/v27/luxburg12a/luxburg12a.pdf