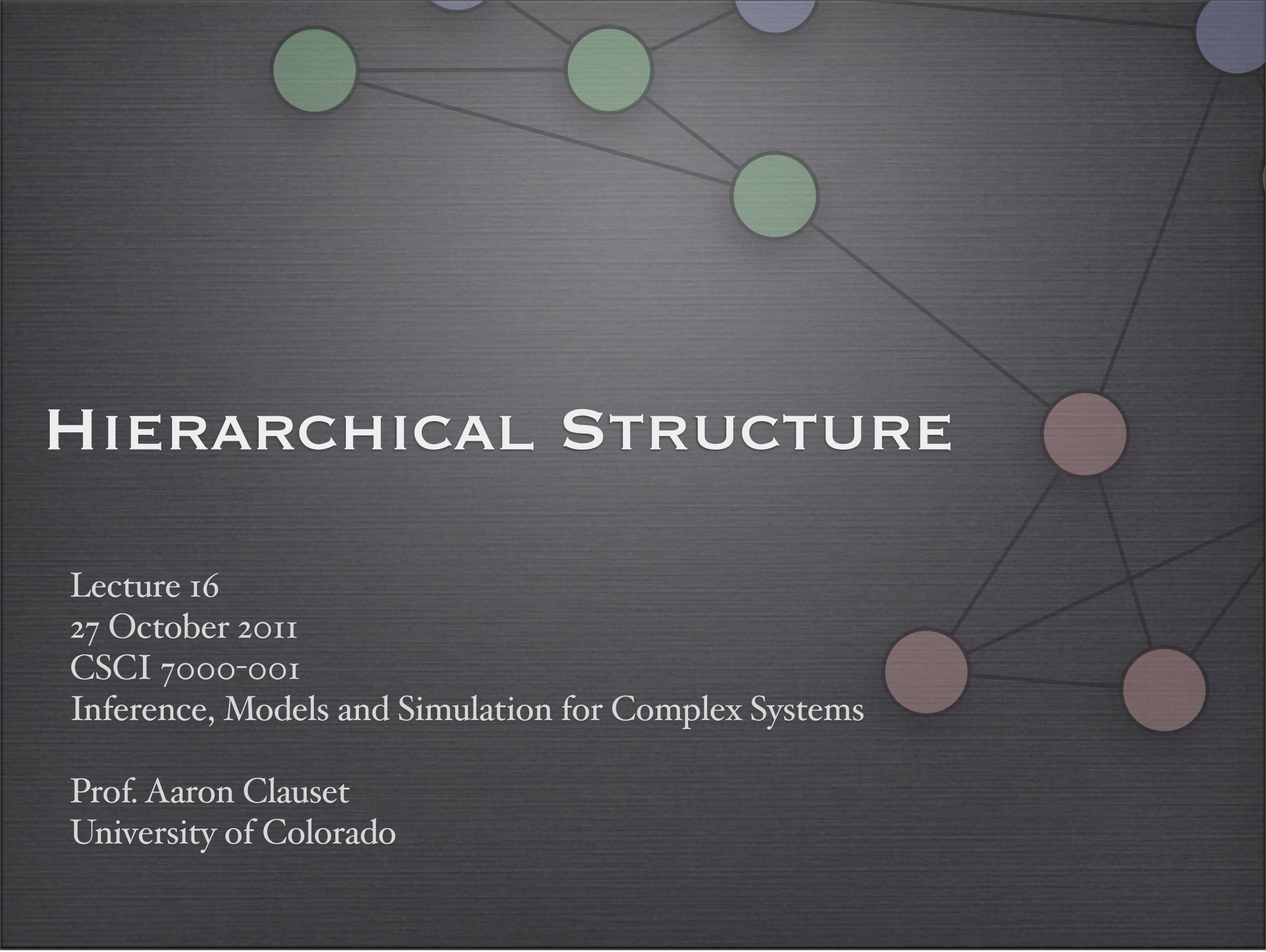
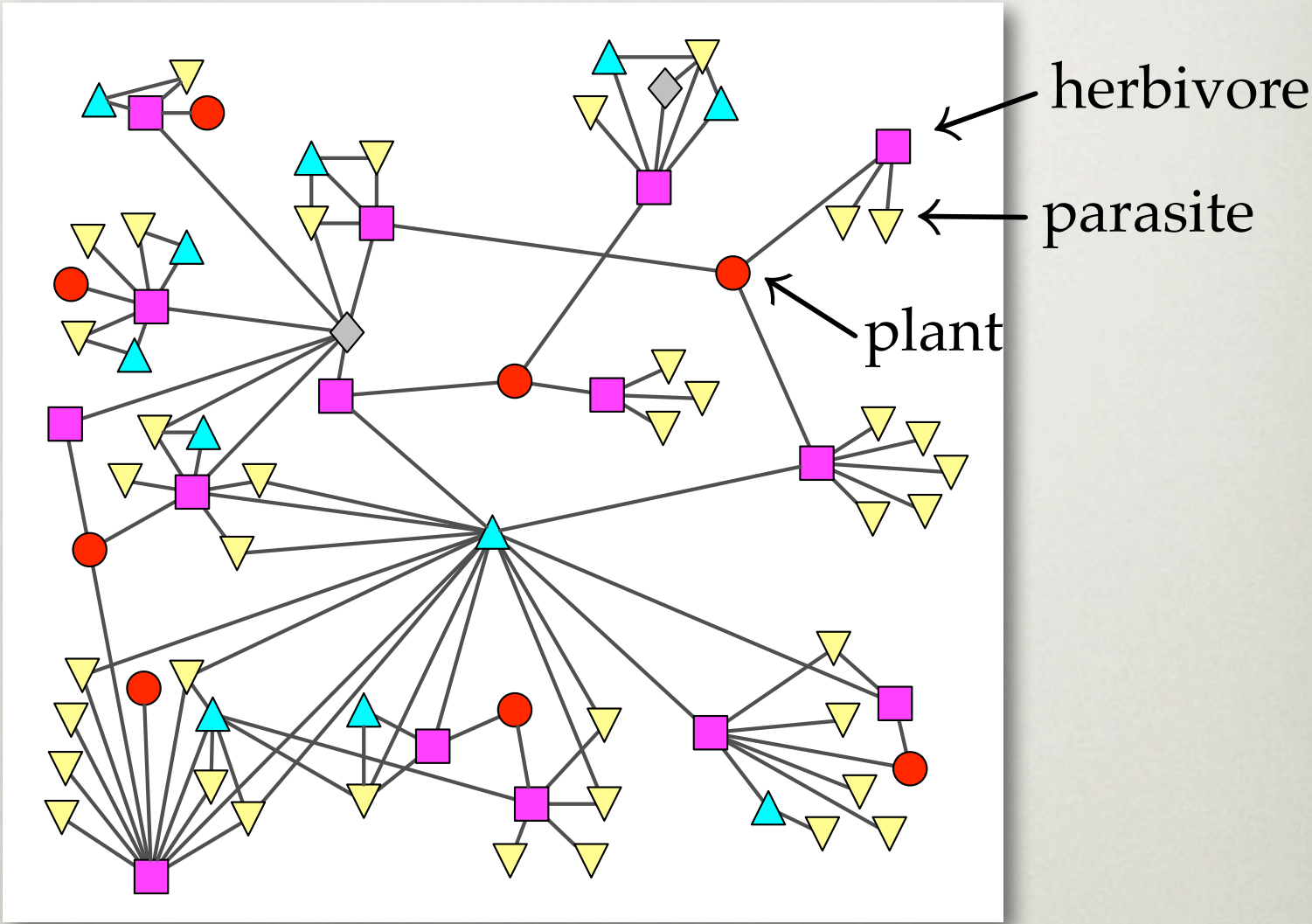


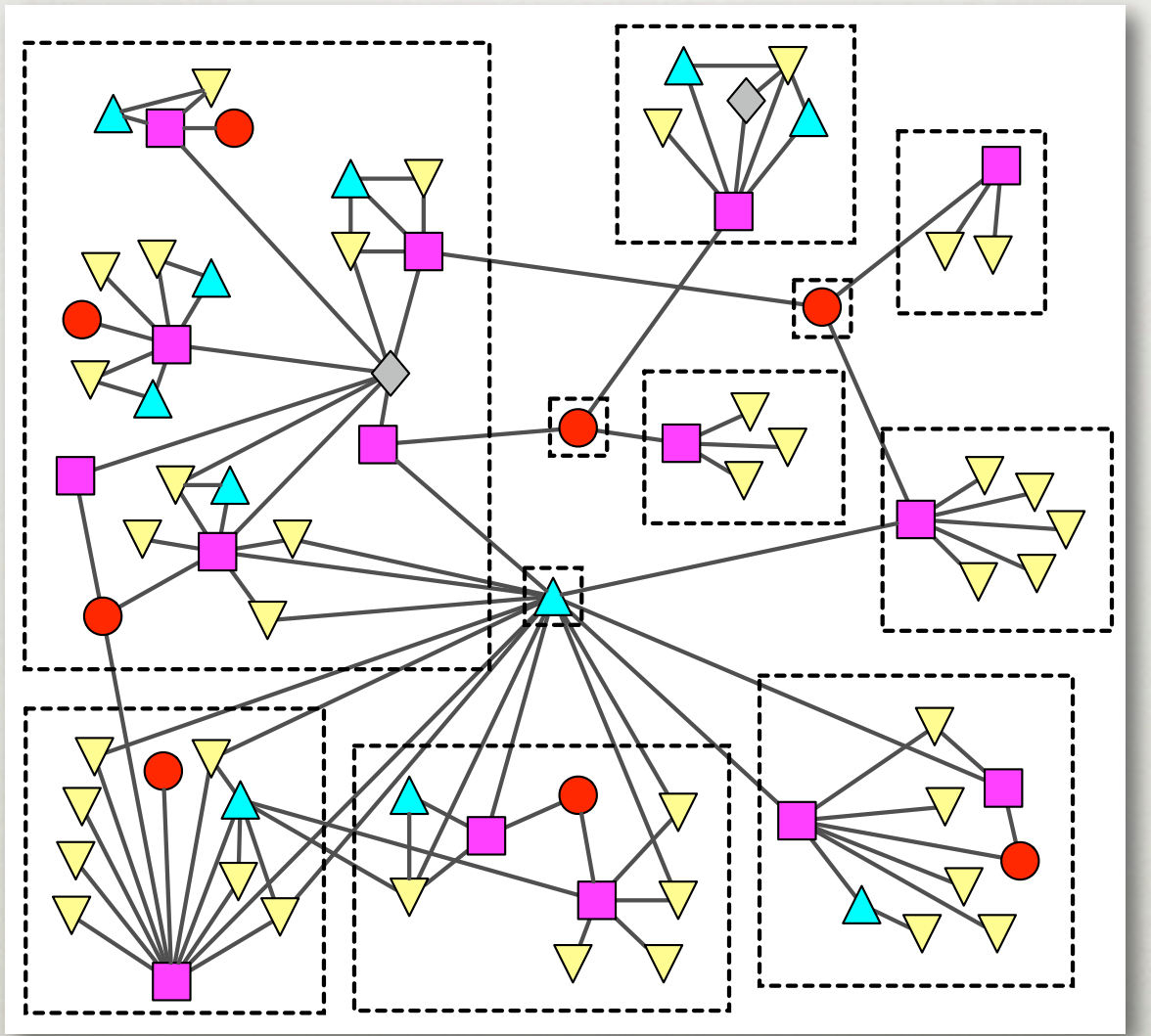
Hierarchical Structure

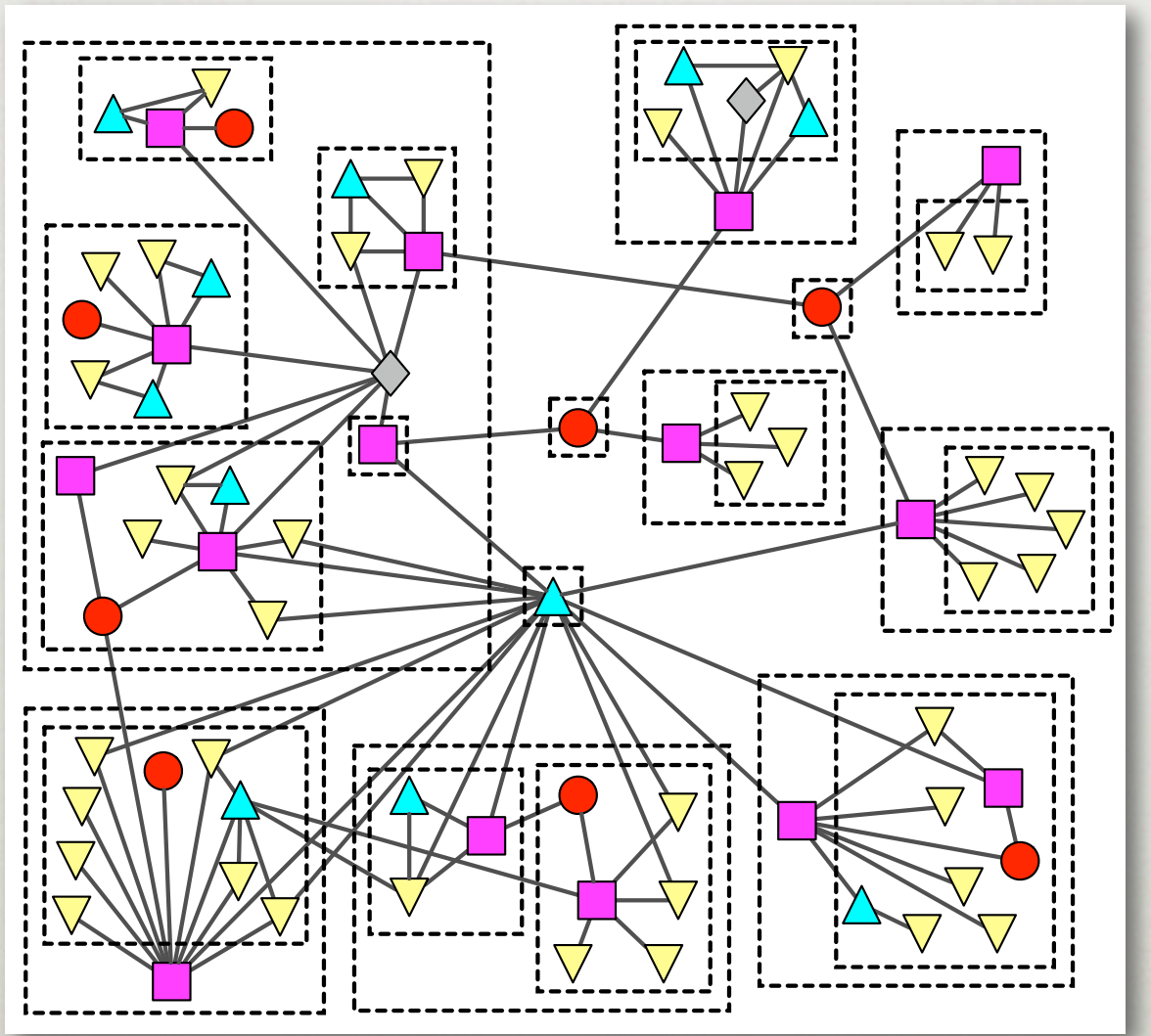
Lecture 16
27 October 2011
CSCI 7000-001
Inference, Models and Simulation for Complex Systems

Prof. Aaron Clauset
University of Colorado

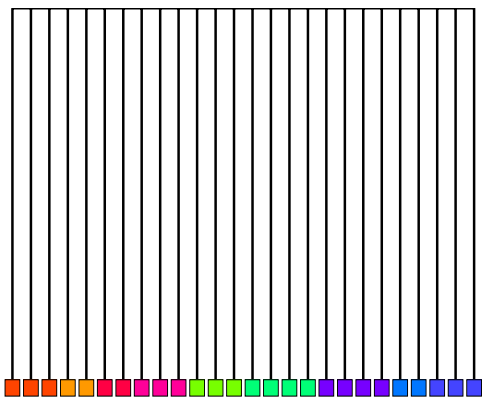


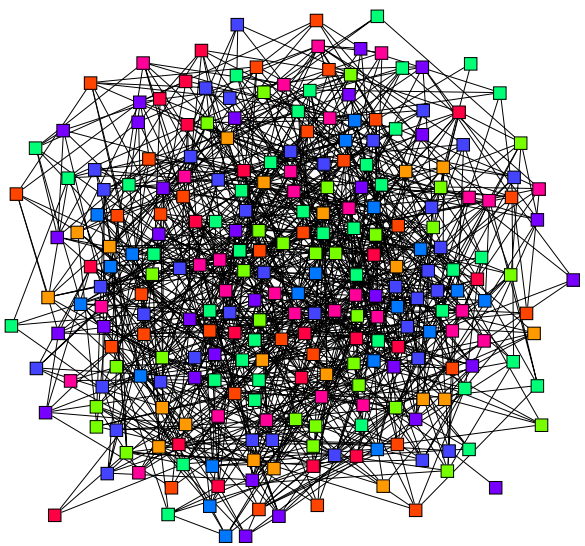
thanks to Jennifer Dunne for the network data





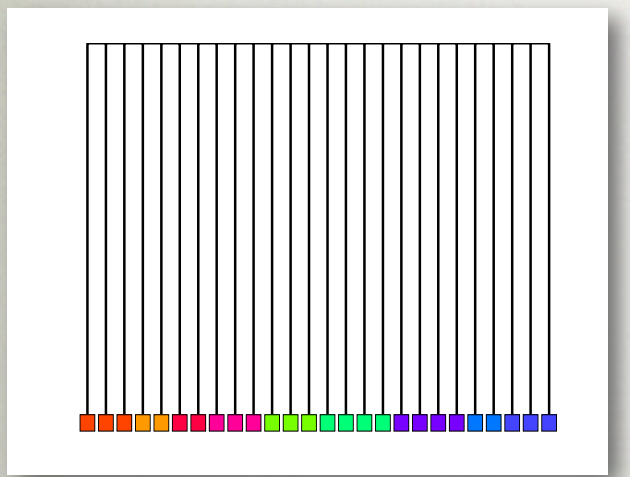
no structure

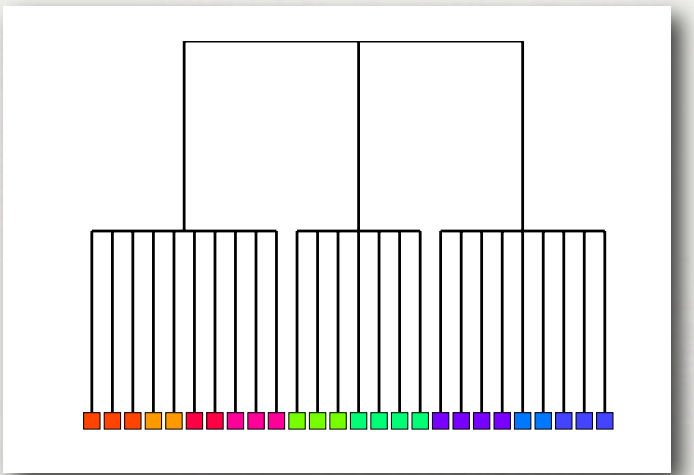


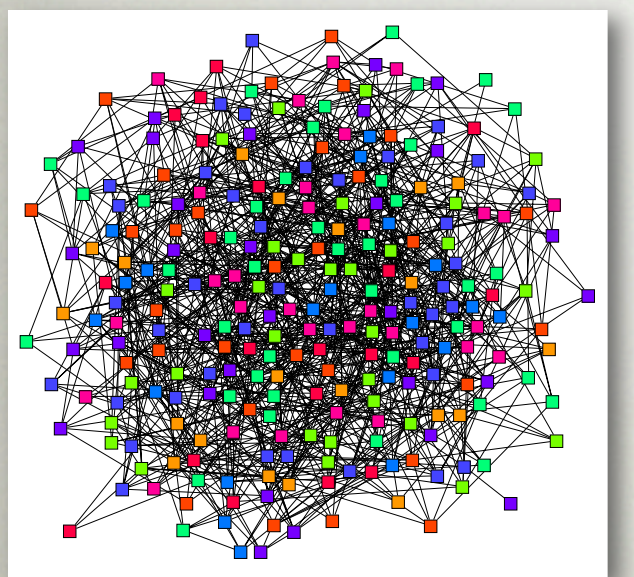


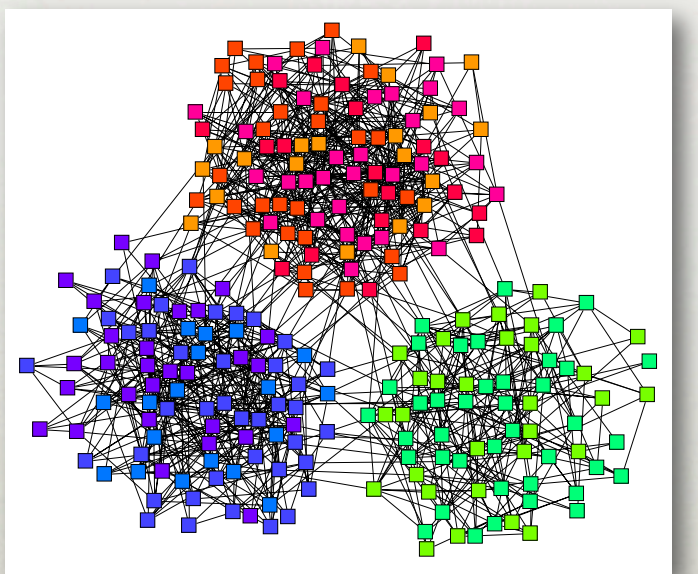
no structure

modular structure



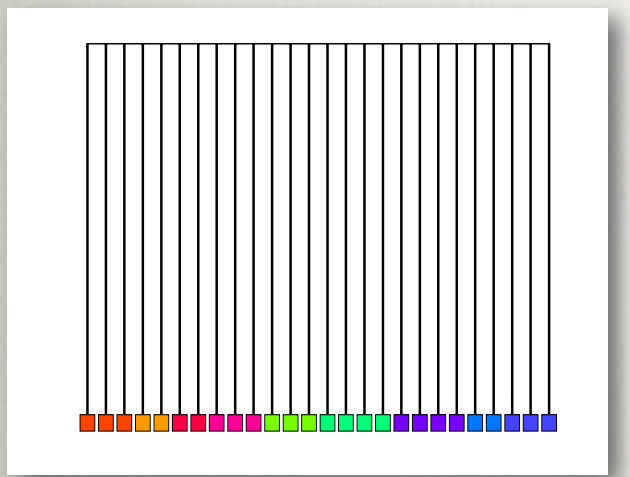




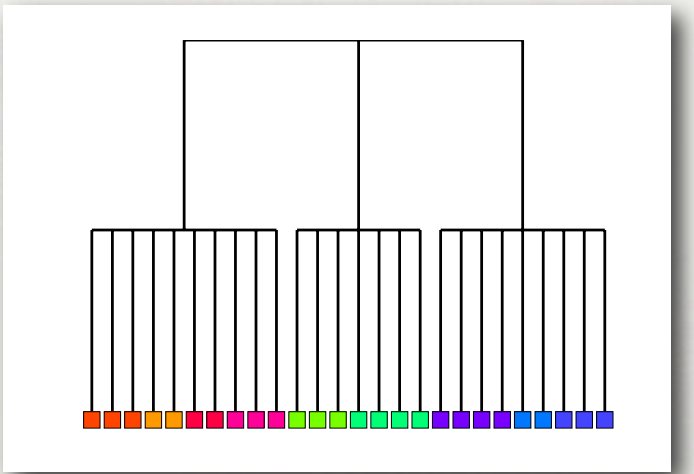


one scale

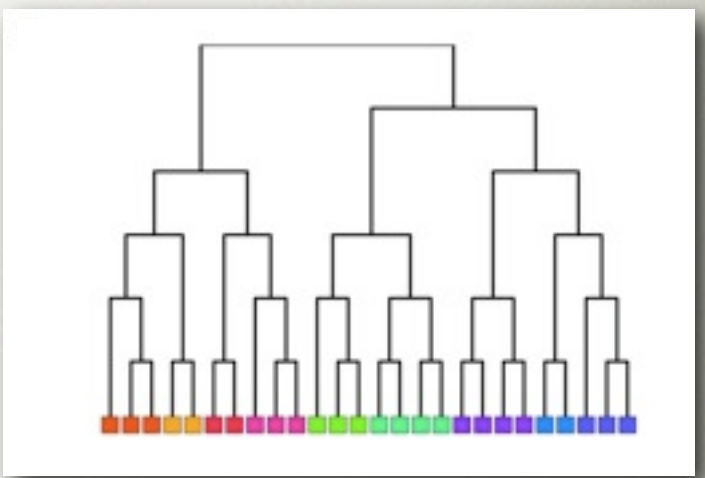
no structure

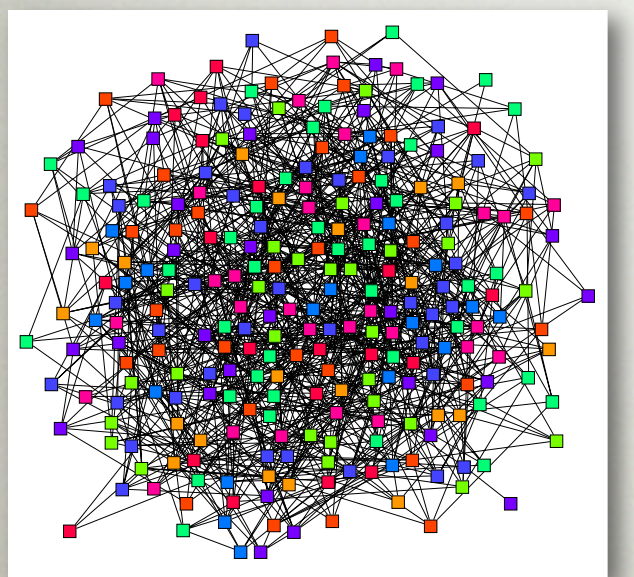


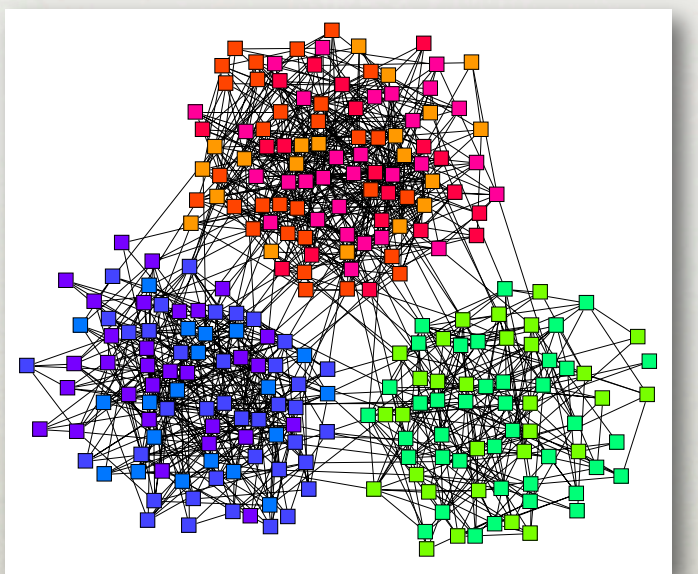
modular structure



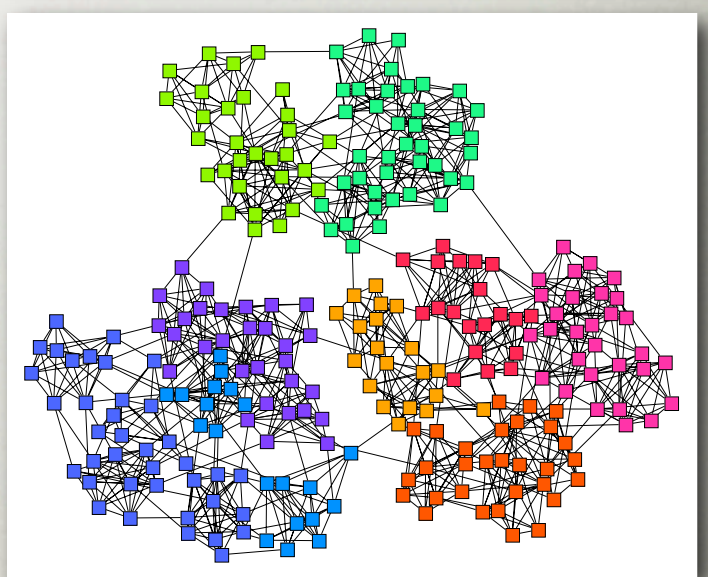
hierarchical structure







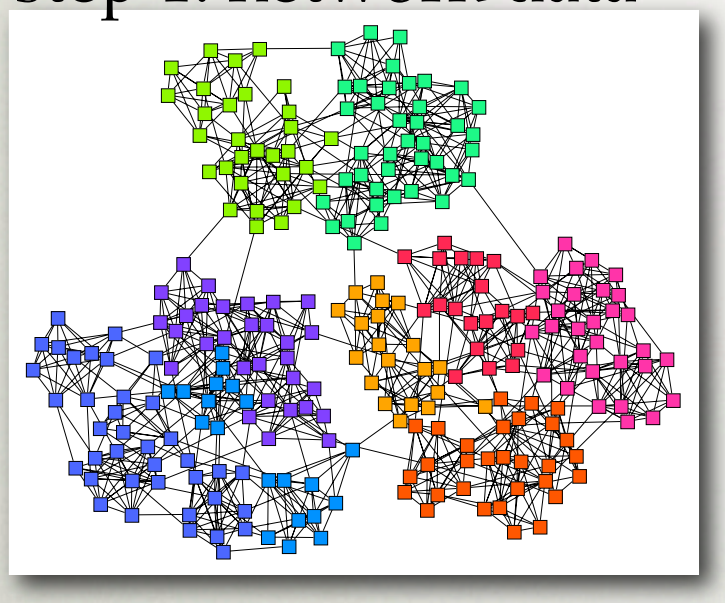
one scale



multi-scale

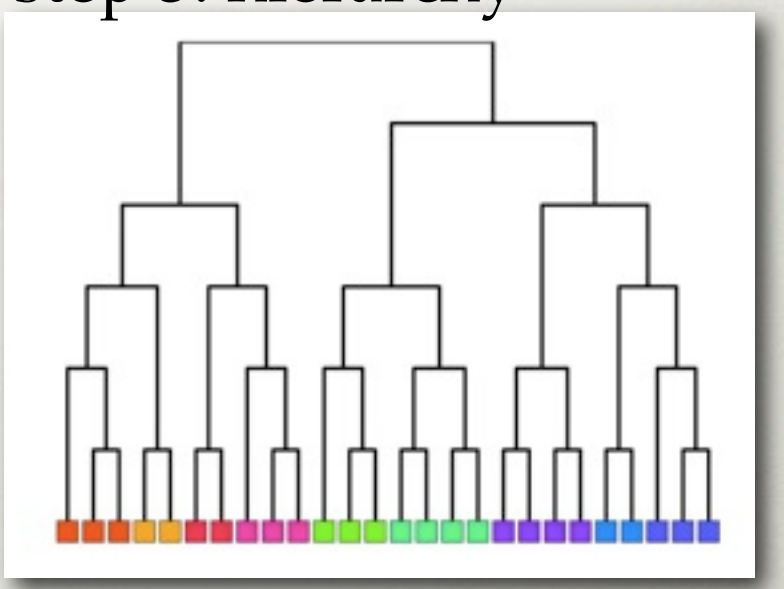
how can we measure a network's hierarchy?

step 1: network data



?

step 3: hierarchy

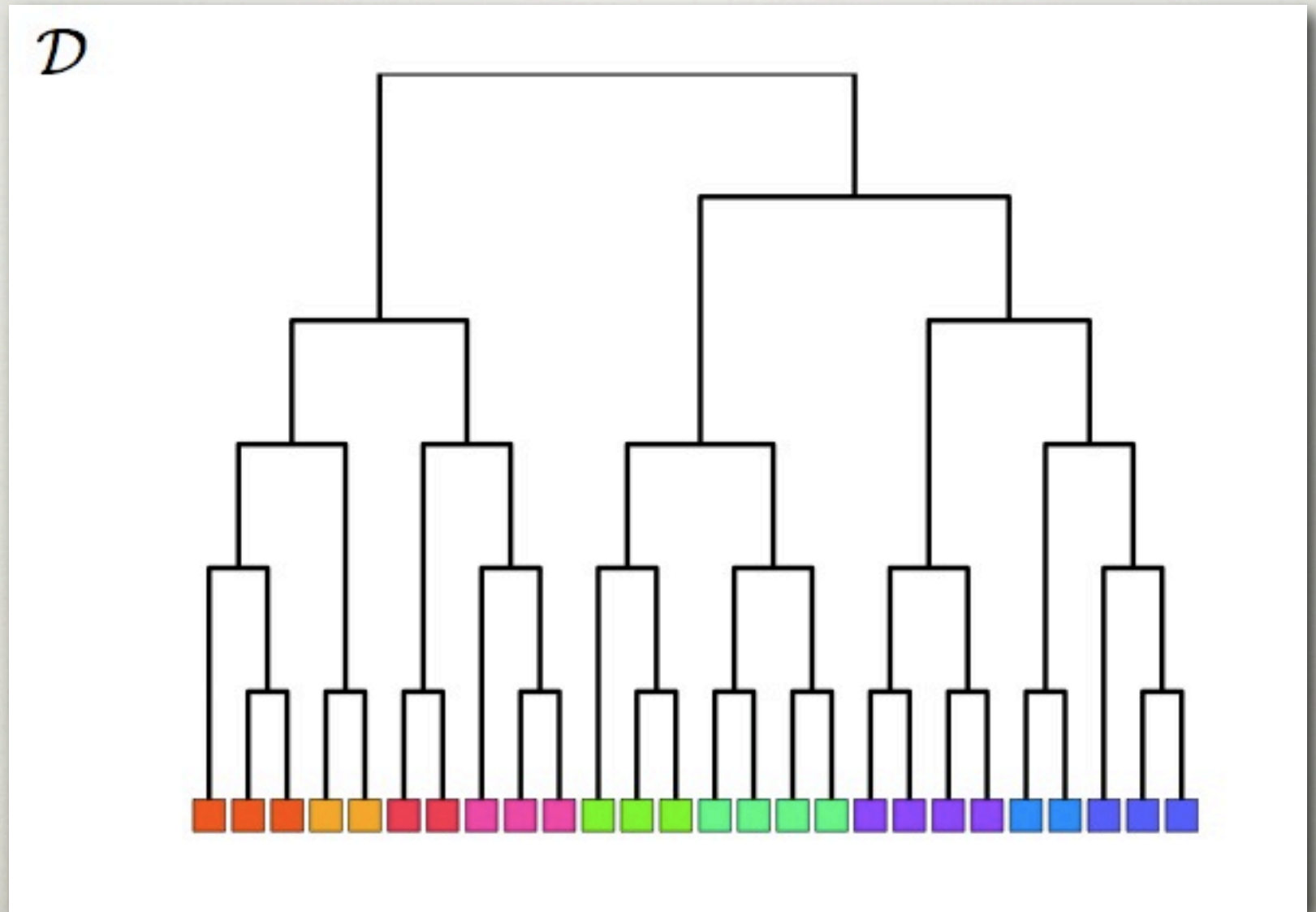


One Approach

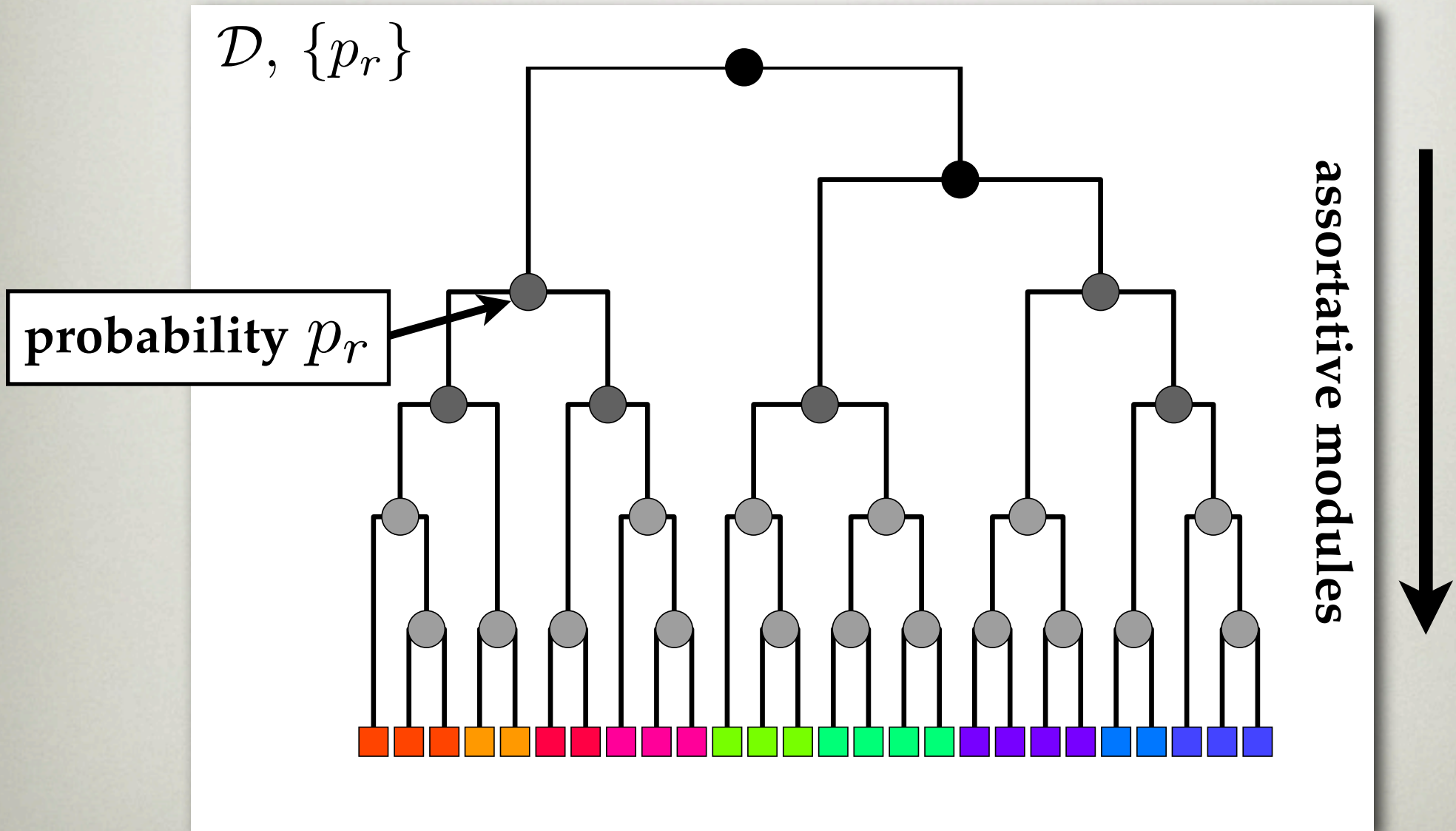
model-based inference

1. describe how to generate hierarchies (a model)
2. estimate / learn model from data (algorithms)
3. test fitted model(s)
4. extract predictions, insight

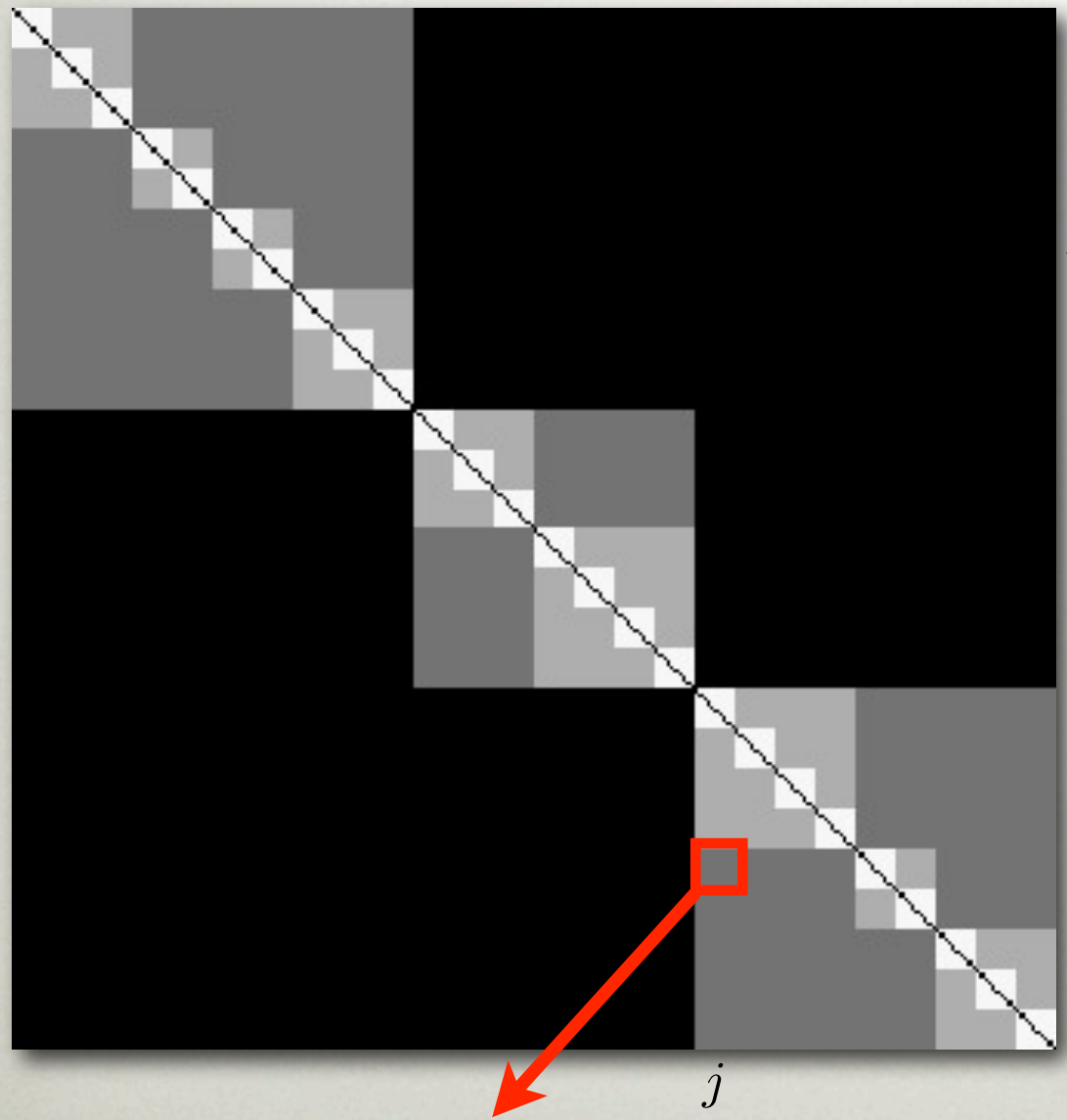
A Model of Hierarchy



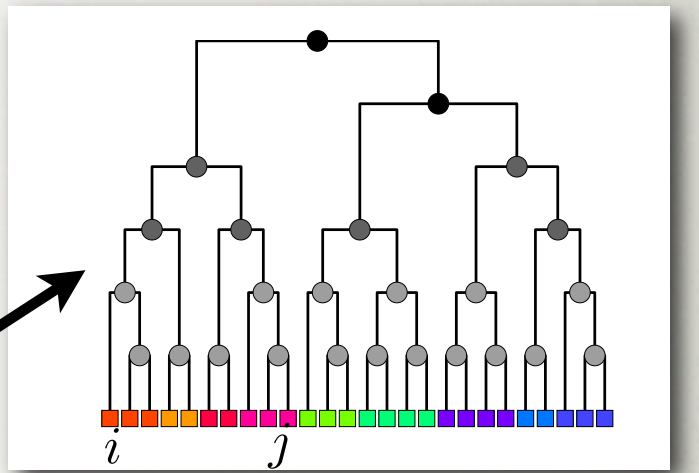
A Model of Hierarchy



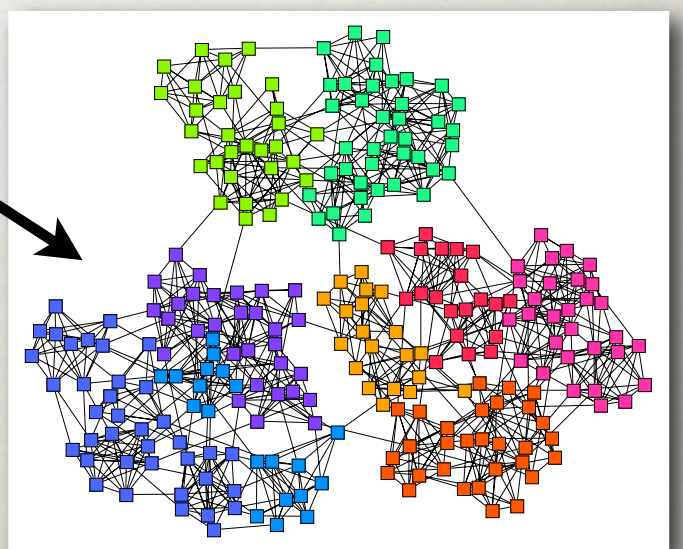
"inhomogeneous" random graph



model



instance



$$\Pr(i,j \text{ connected}) = p_r$$
$$= p_{(\text{lowest common ancestor of } i,j)}$$

Hierarchical Random Graph

- explicit model = explicit assumptions
- flexible (2n parameters)
- captures structure at all scales
- mixtures of assortativity, disassortativity
- decomposition into set of random bipartite graphs
- learnable directly from data

Learning From Data

a direct approach

- **likelihood function** $\mathcal{L} = \Pr(\text{ data} \mid \text{model})$

  ($\mathcal{L}$ scores **quality** of model)

- **sample all good models**

  via Markov chain Monte Carlo*
  over all dendrograms

- **technical details in**

  Clauset, Moore and Newman, *Nature* **453**, 98-101 (2008) and

  Clauset, Moore and Newman, *ICML* (2006)

* other sampling or optimization methods possible

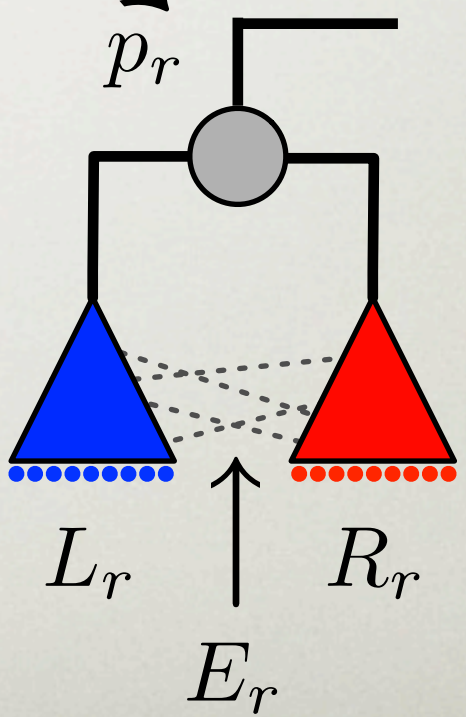
Likelihood Function

$$\mathcal{L}(\mathcal{D}, \{p_r\}) = \prod_r p_r^{E_r} (1 - p_r)^{L_r R_r - E_r}$$

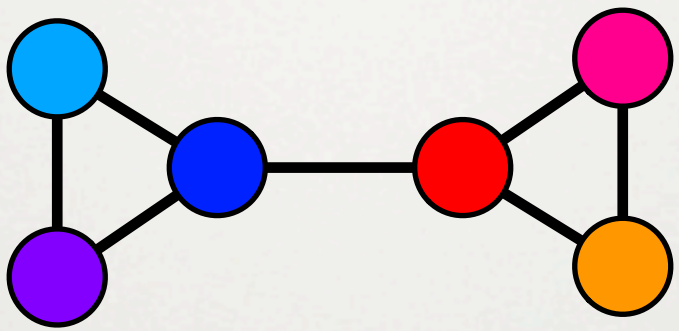
L_r = number nodes in left subtree

R_r = number nodes in right subtree

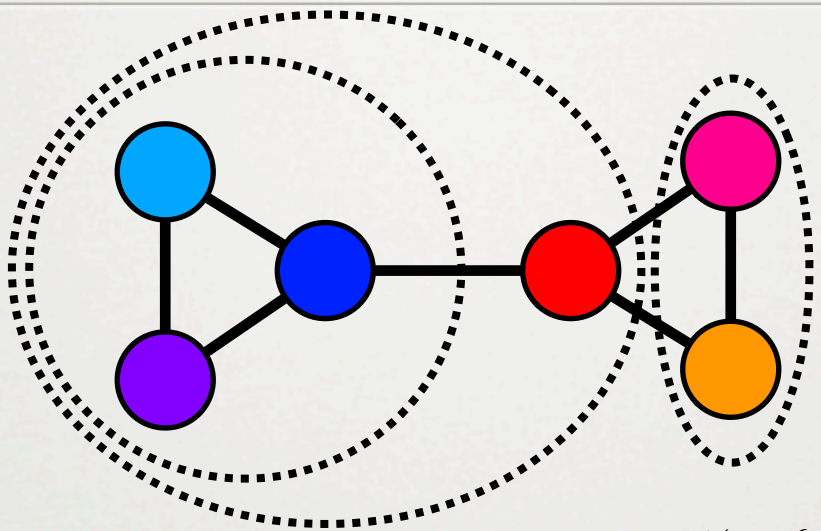
E_r = number edges with r as lowest common ancestor



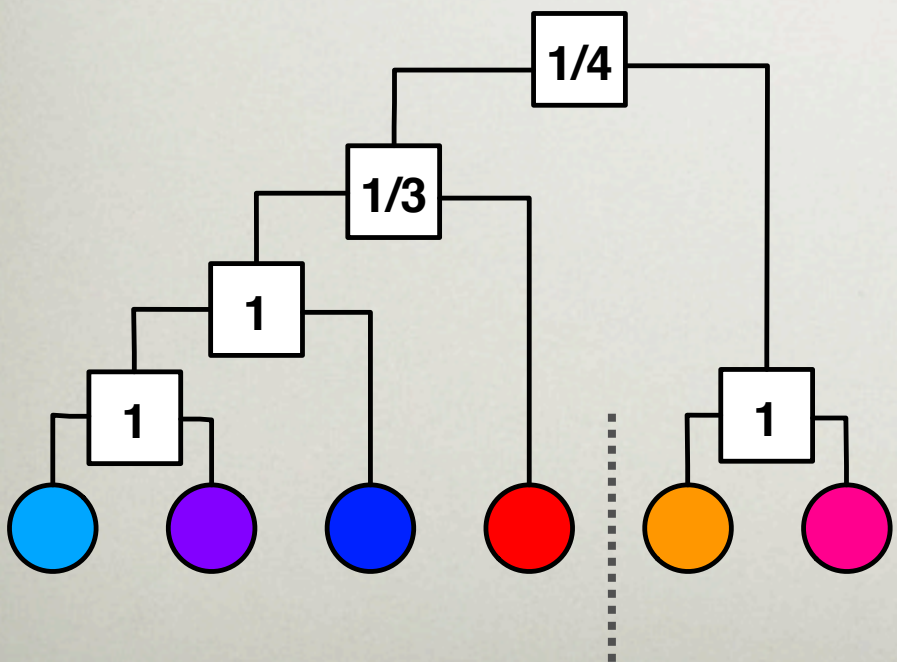
Example



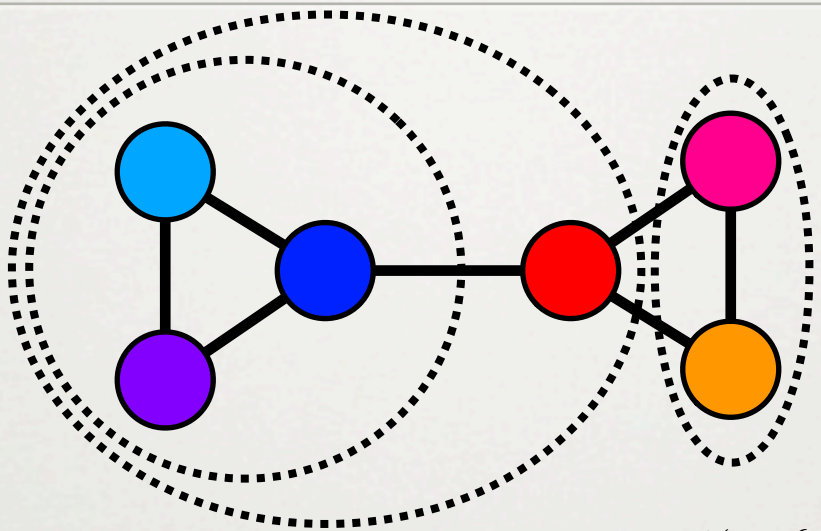
Bad Dendrogram

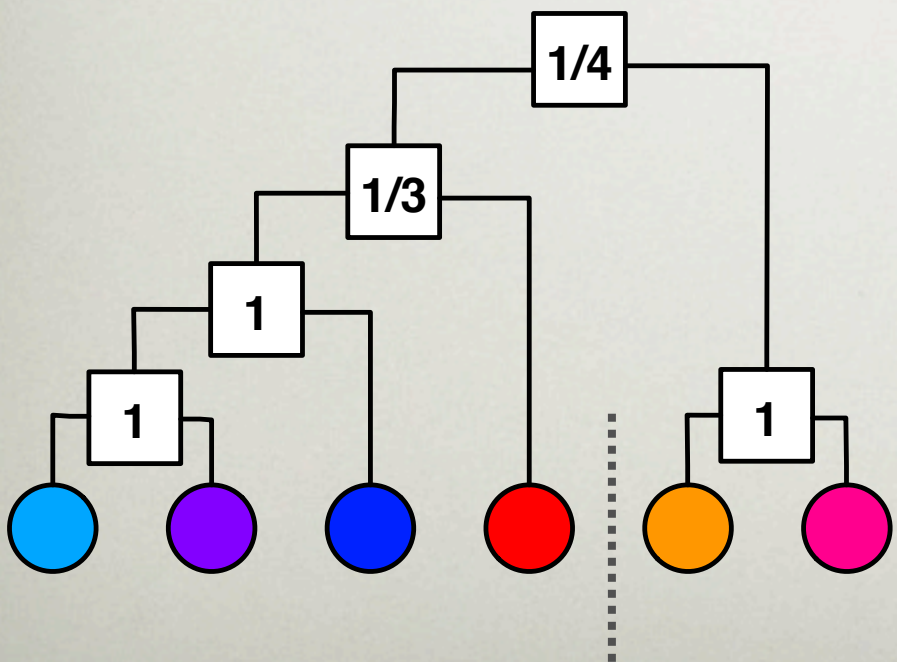


$$\mathcal{L}(\mathcal{D}, \{p_r\}) = \prod_r p_r^{E_r} (1 - p_r)^{L_r R_r - E_r}$$



Bad Dendrogram



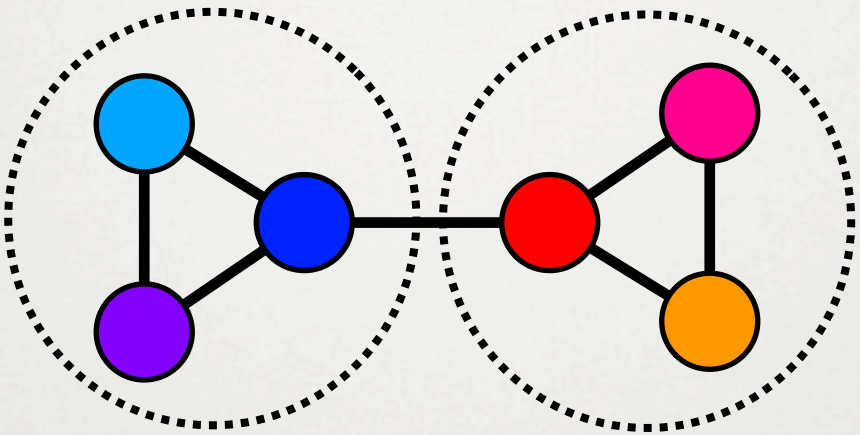


$$\mathcal{L}(\mathcal{D}, \{p_r\}) = \prod_r p_r^{E_r} (1 - p_r)^{L_r R_r - E_r}$$

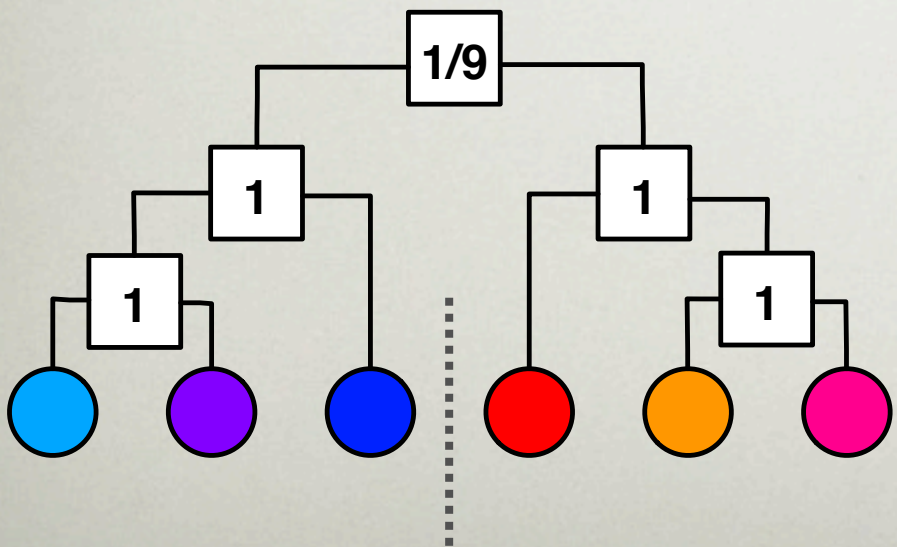
$$\mathcal{L} = \left[\left(\frac{1}{3}\right)^1 \left(\frac{2}{3}\right)^2 \right] \cdot \left[\left(\frac{1}{4}\right)^2 \left(\frac{3}{4}\right)^6 \right]$$

$$\mathcal{L} = 0.0016$$

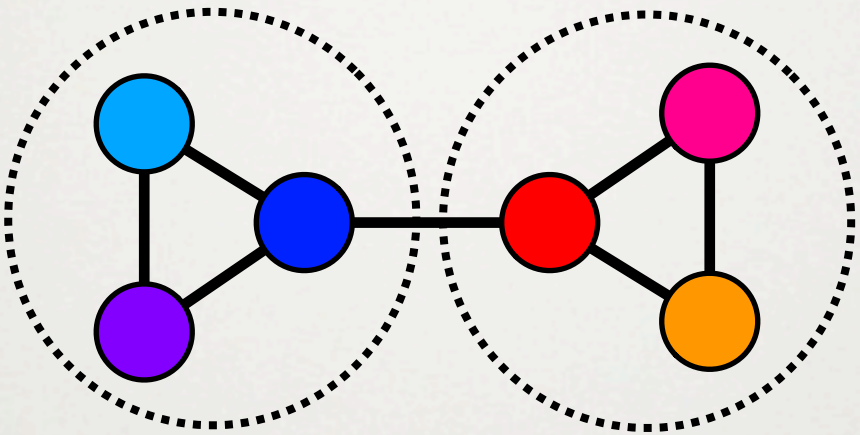
Good Dendrogram



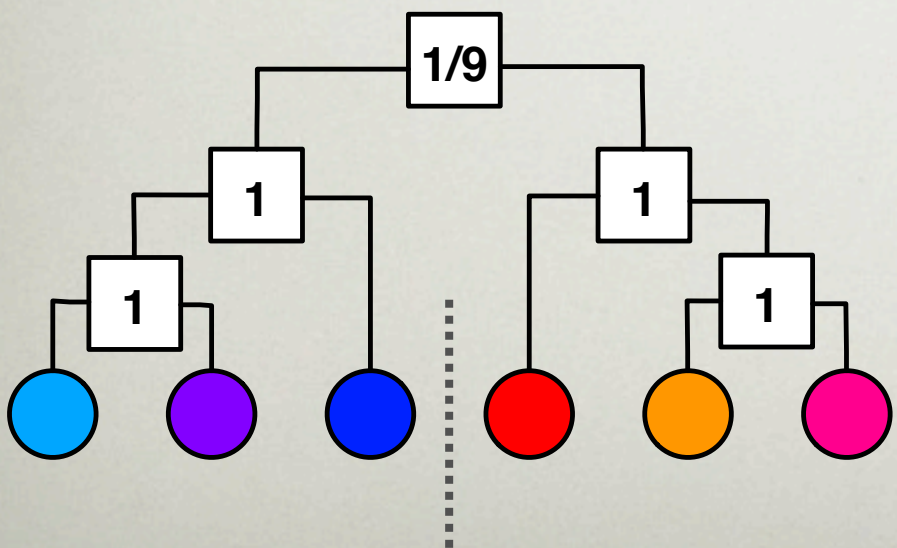
$$\mathcal{L}(\mathcal{D}, \{p_r\}) = \prod_r p_r^{E_r} (1 - p_r)^{L_r R_r - E_r}$$



Good Dendrogram



$$\mathcal{L}(\mathcal{D}, \{p_r\}) = \prod_r p_r^{E_r} \left(1 - p_r\right)^{L_r R_r - E_r}$$



$$\mathcal{L} = \left[\left(\frac{1}{9}\right)^1 \left(\frac{8}{9}\right)^8 \right]$$

$$\mathcal{L} = 0.0433$$

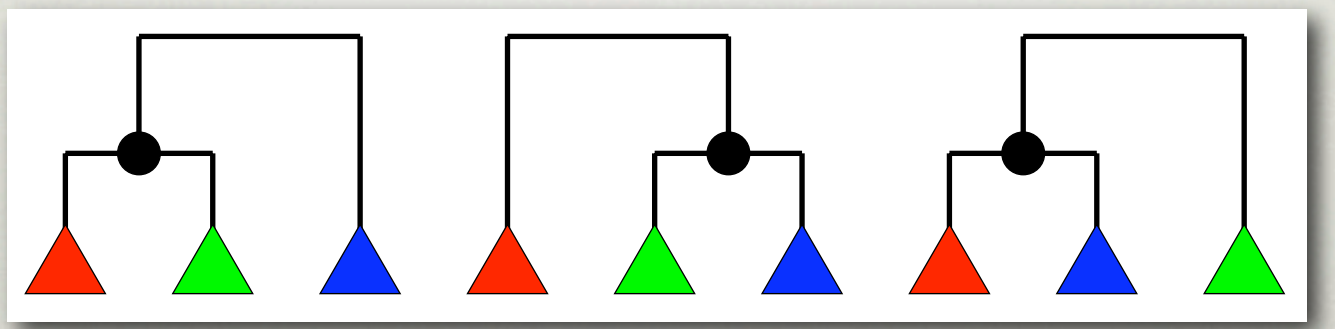
Markov chain Monte Carlo (MCMC)

Given $\mathcal{D}$, choose random internal node

Choose random reconfiguration of subtrees [ergodicity]

Recompute probabilities $\{p_r\}$ and likelihood $\mathcal{L}$

Sampling states according to their likelihood [detailed balance]



three subtree configurations

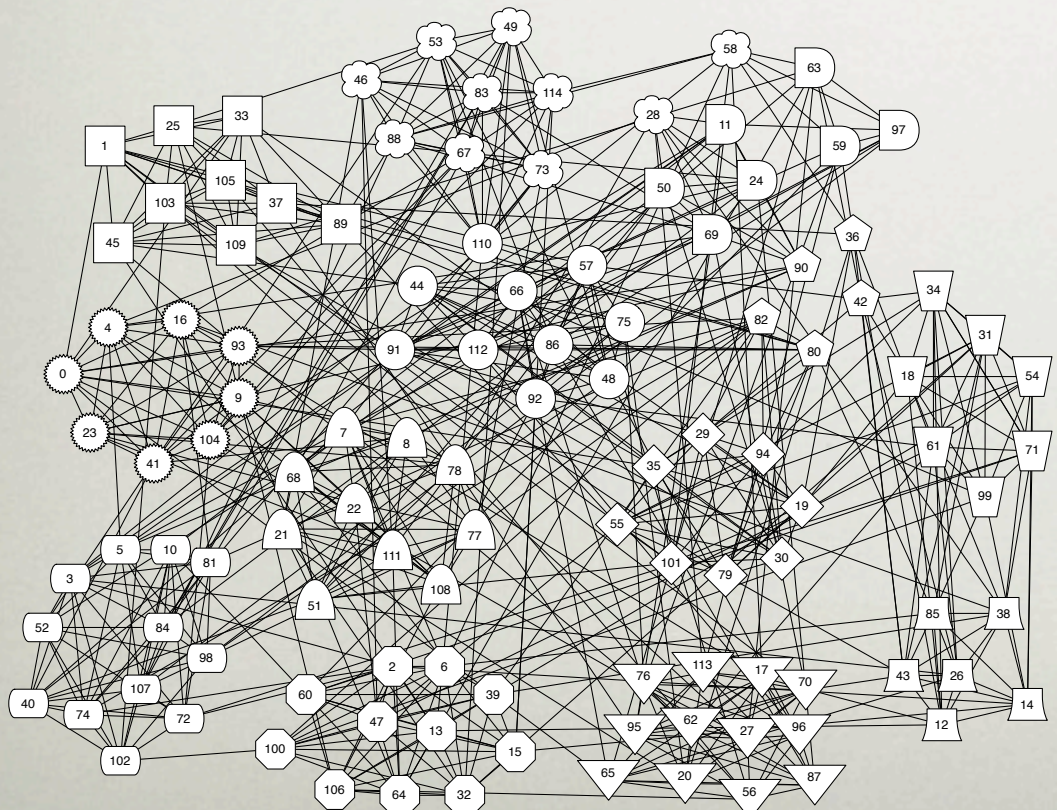
(up to relabeling)

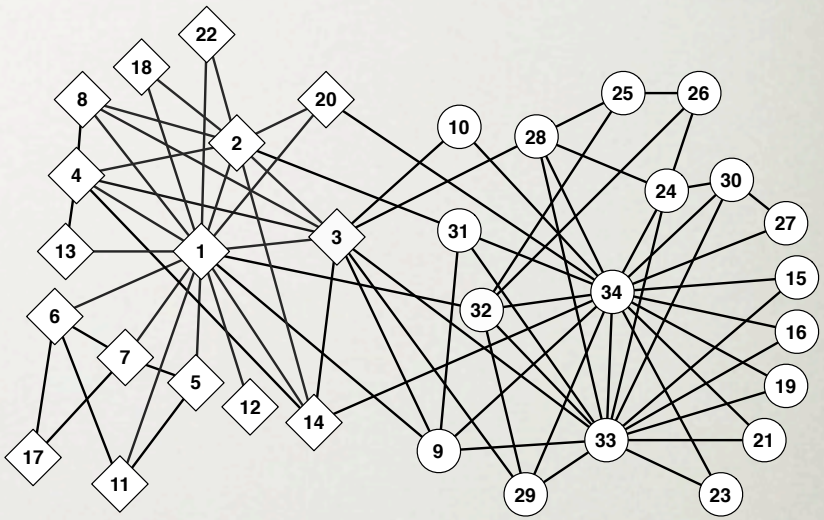
Some applications

Two Case Studies

NCAA Schedule 2000

$n = 115 \quad m = 613$





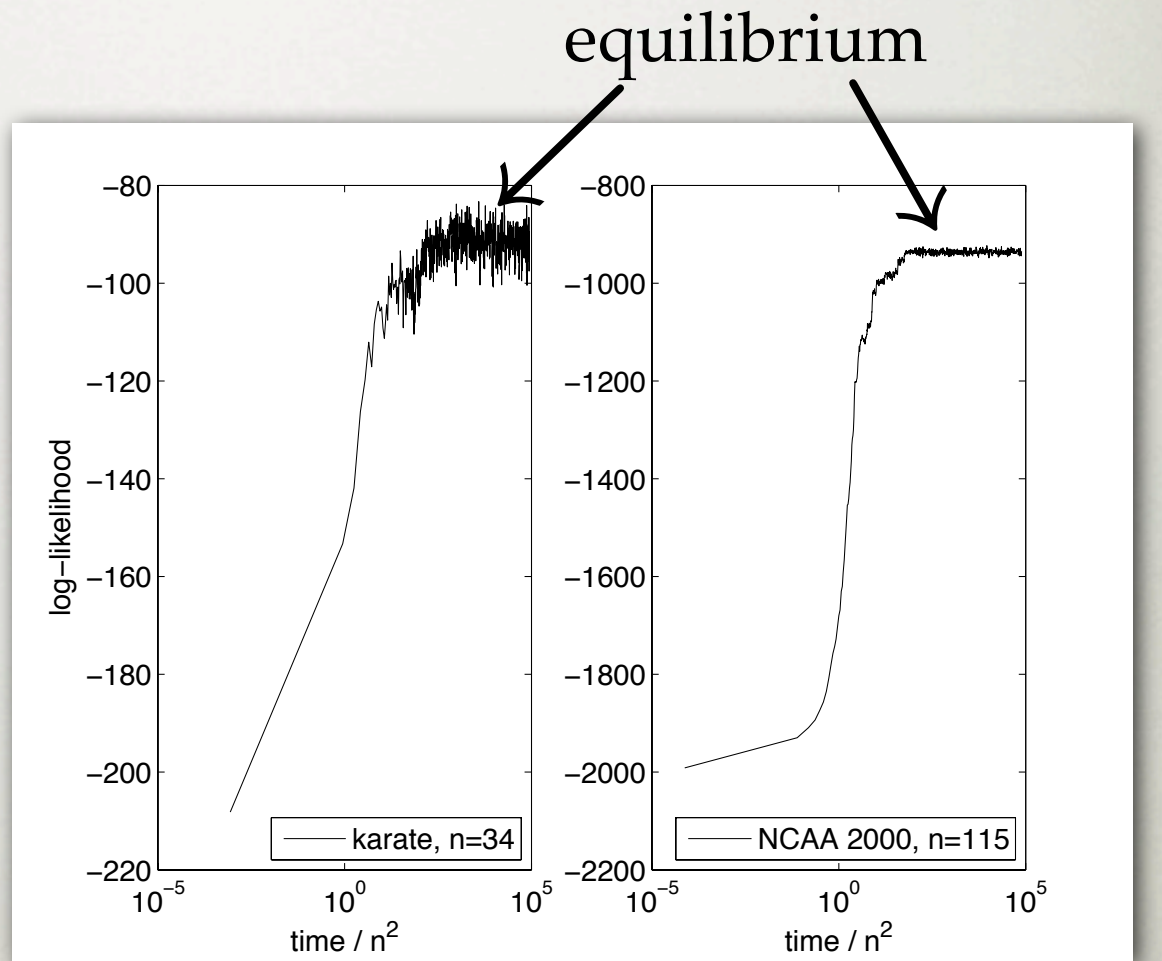
Zachary's Karate Club

$n = 34 \quad m = 78$

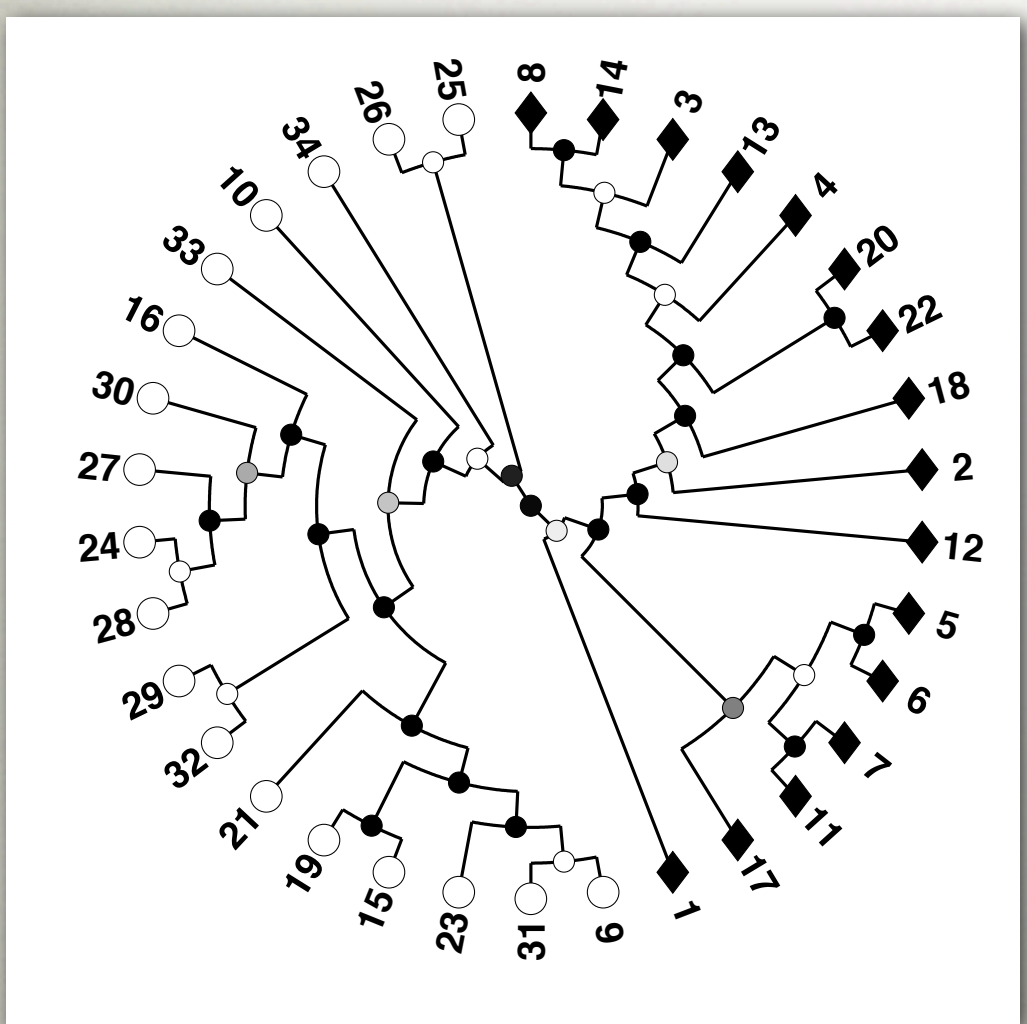
Mixing Times

MCMC mixes relatively quickly

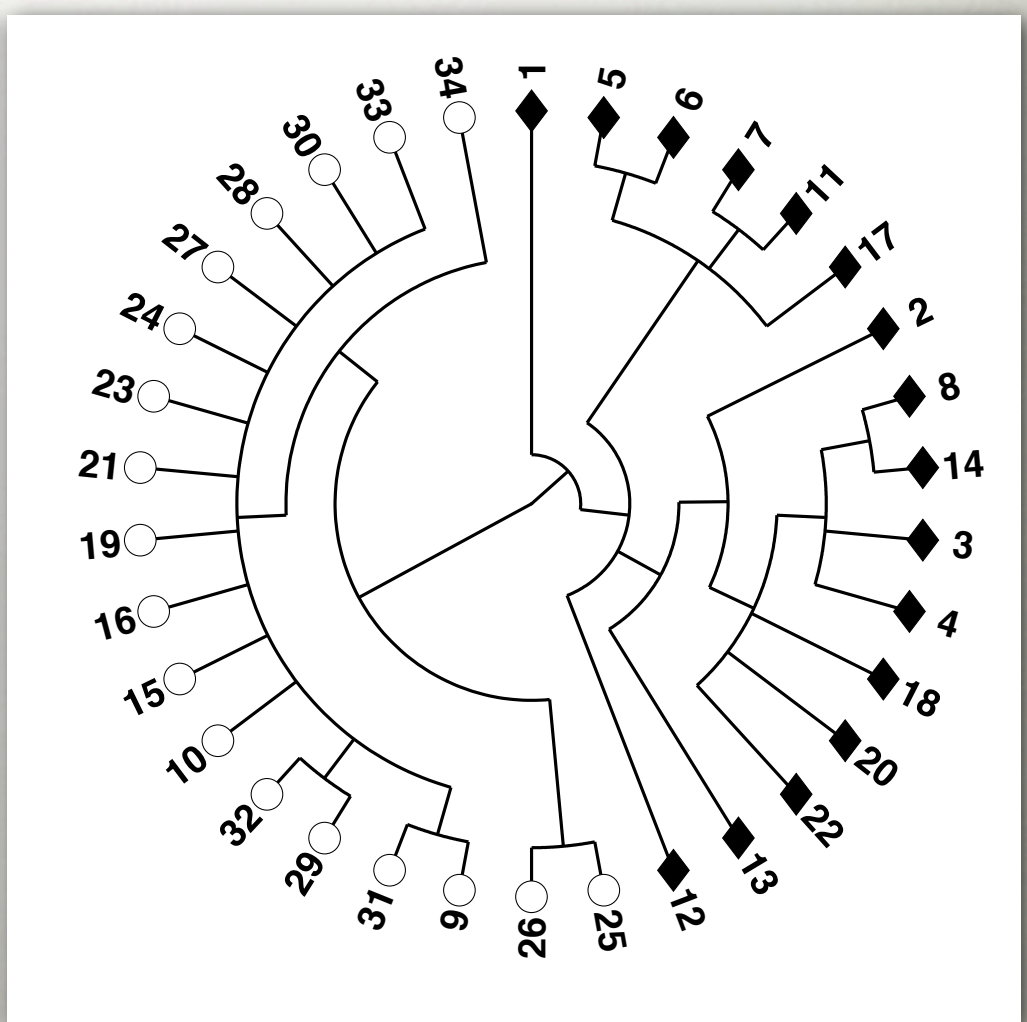
Equilibrium in $\sim O(n^2)$ steps



Hierarchies

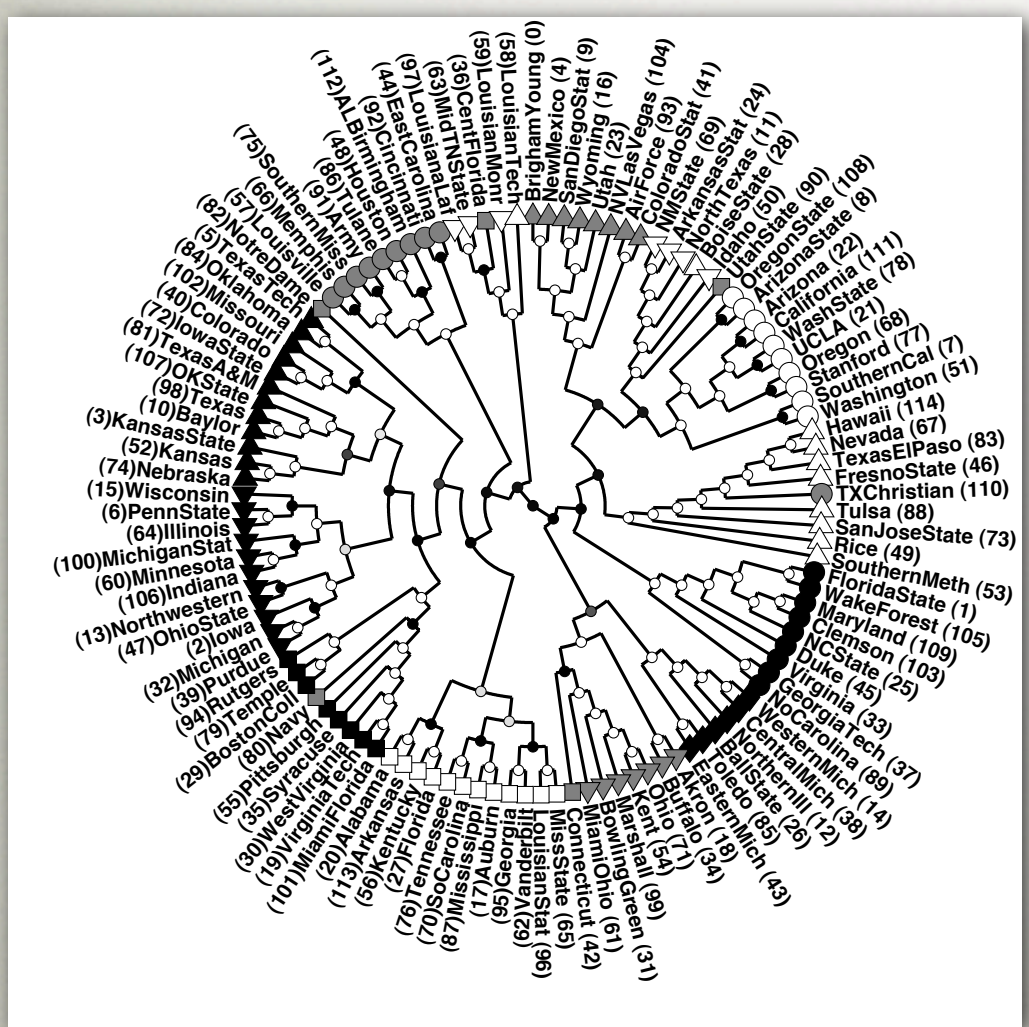


point estimate

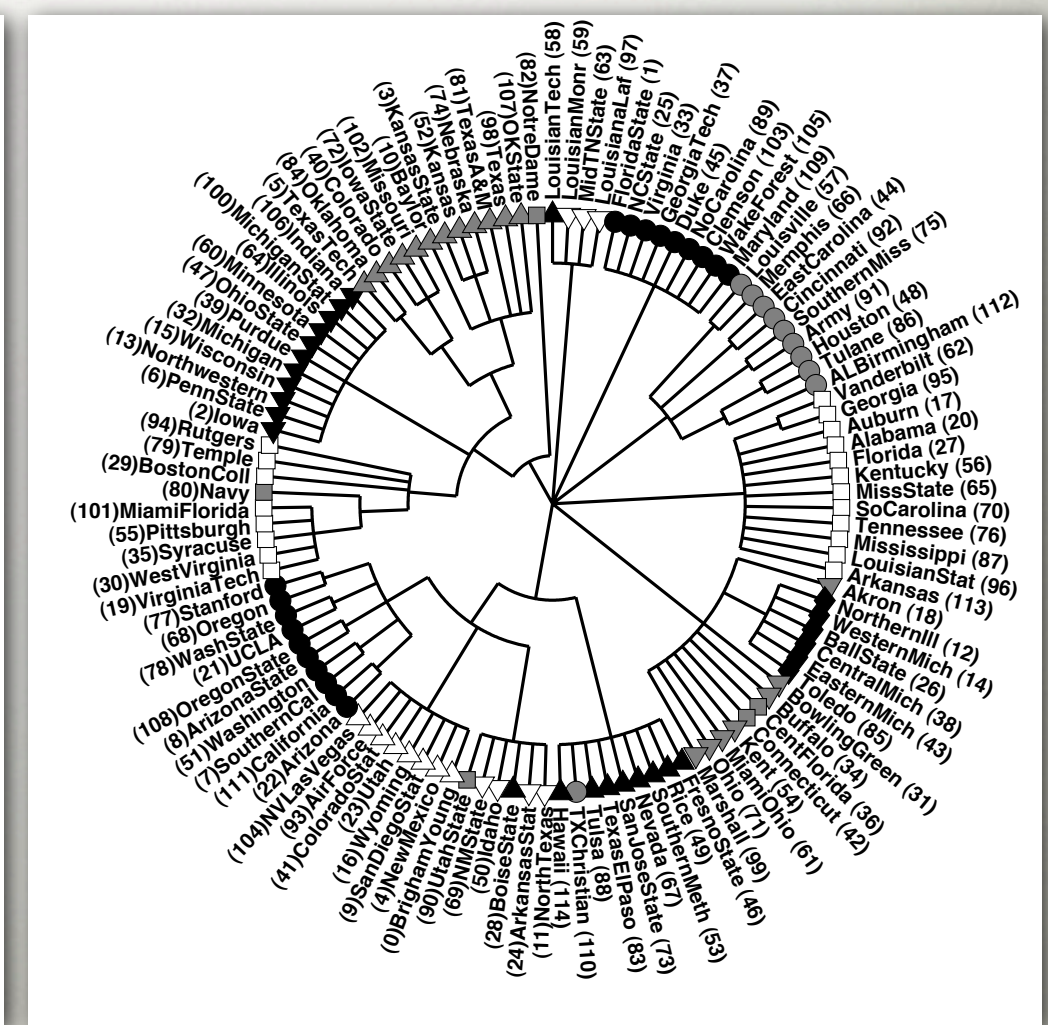


consensus hierarchy

Hierarchies



point estimate



consensus hierarchy

Edge Annotations

Average likelihood of edge existing

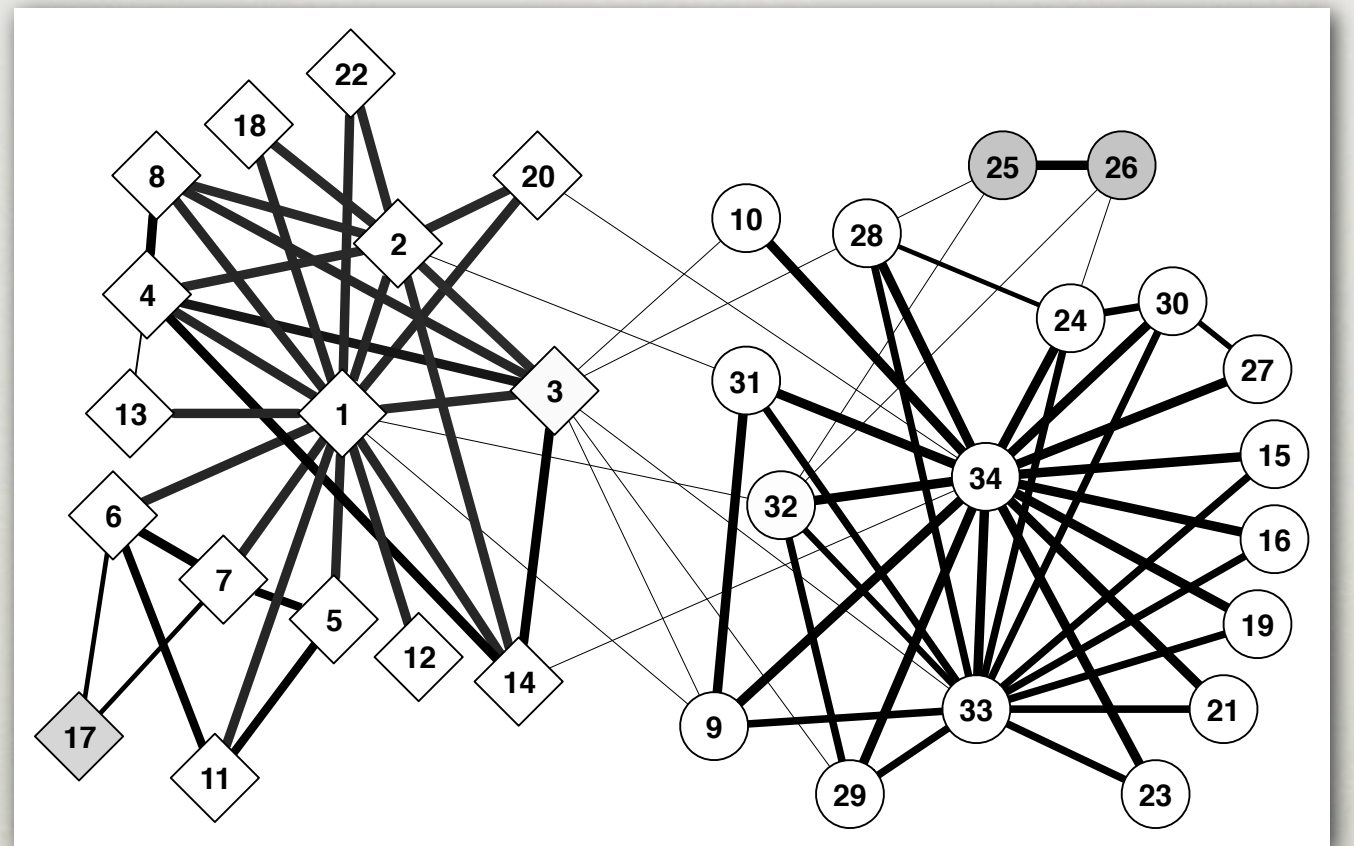
- For each edge (i, j) in G, compute average associated parameter $\langle \theta_r \rangle_{(i,j)}$ over sampled models
- $\langle \theta_r \rangle_{(i,j)}$ is edge annotation (weight)

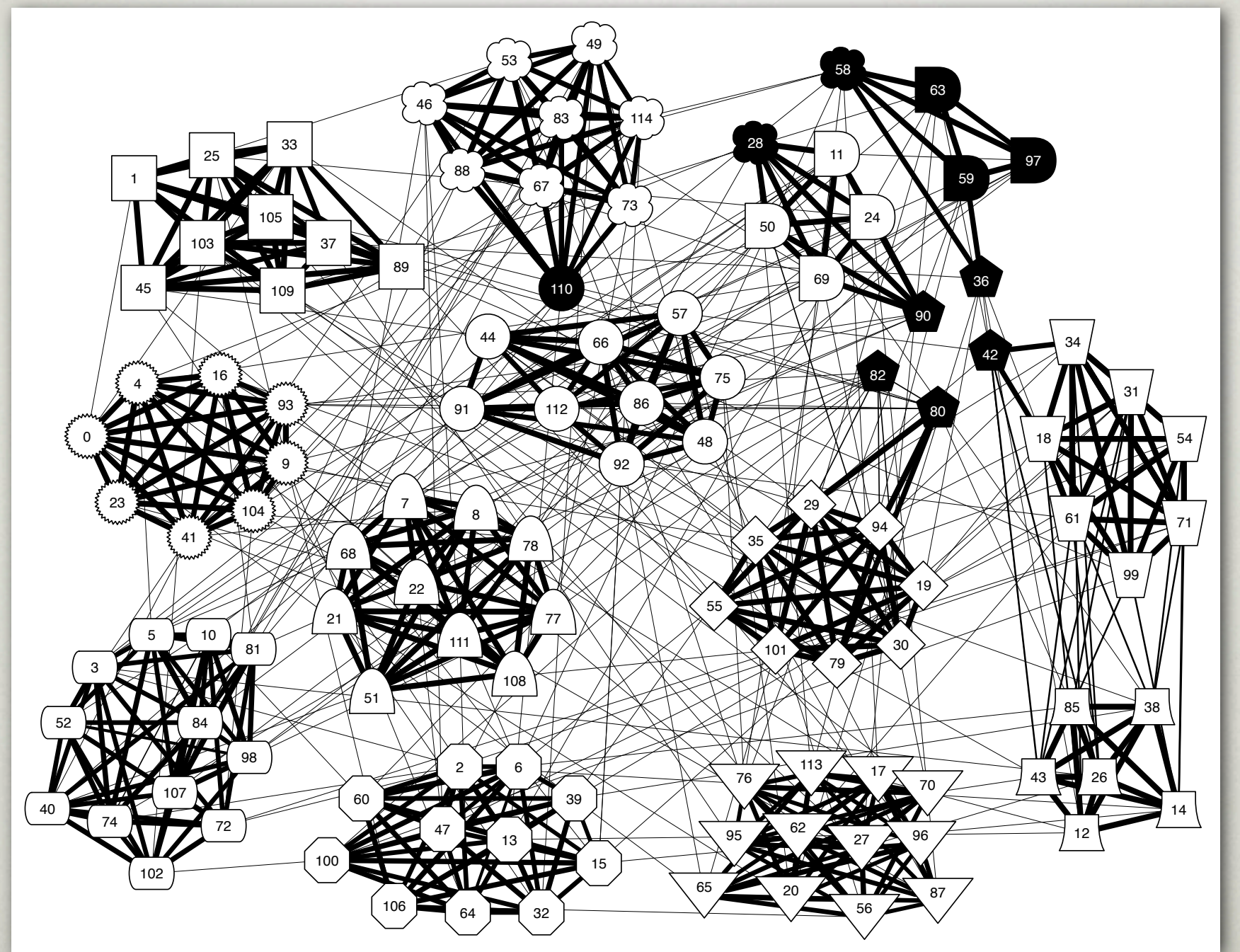
Vertex Annotations

Group-affiliation strengths

- If each vertex has known group label
- Ask, how often does vertex i appear in a subtree with majority of its fellows?
- Frequency is vertex annotation (strength)

Edge, Note Annotations



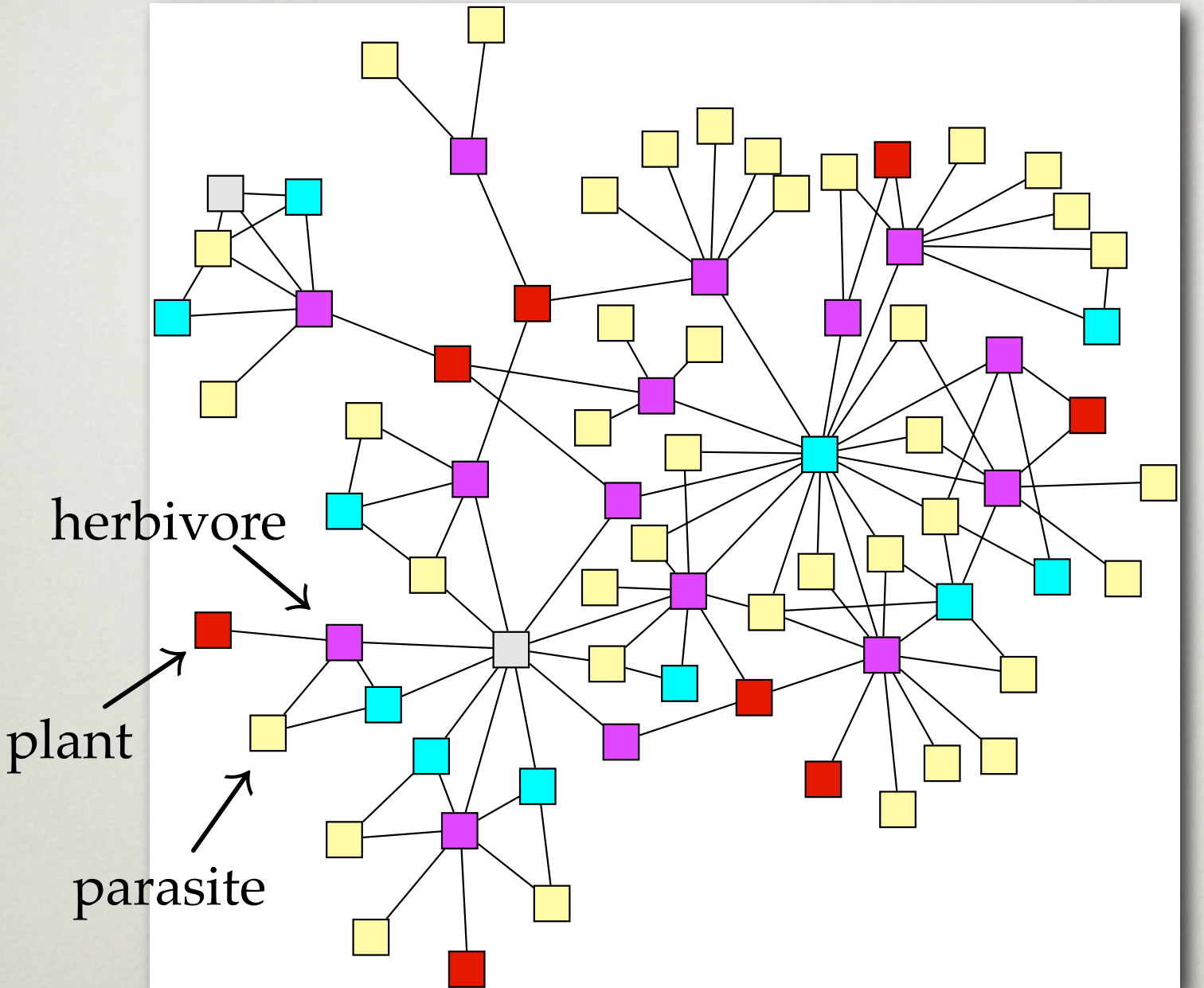

49
53
46
83
114
58
63
28
11
97
59
88
67
73
25
33
1
105
103
37
89
45
109
50
24
69
36
110
90
57
44
66
75
82
42
34
4
16
93
91
112
86
80
31
0
9
48
92
18
54
23
104
41
7
8
29
94
61
71
68
78
35
22
19
99
55
21
77
111
101
79
30
5
10
81
3
108
51
85
38
52
84
98
2
6
76
113
17
70
43
26
40
107
60
39
95
62
27
96
14
74
72
47
13
12
100
15
65
20
87
102
106
64
32
56

From Graph to Ensemble

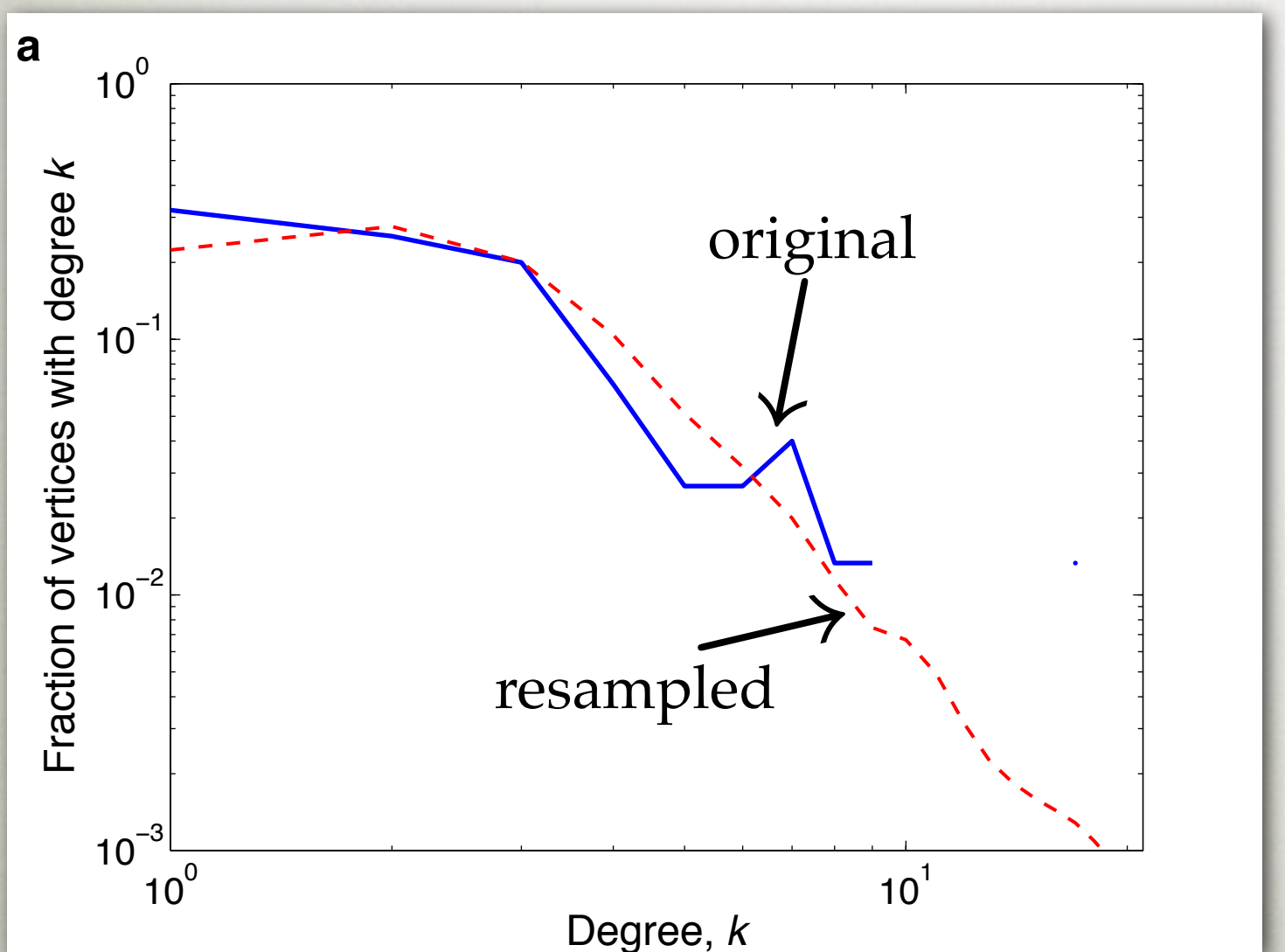
From Graph to Ensemble

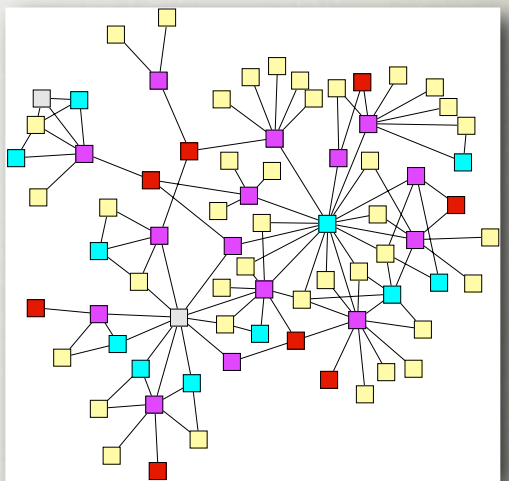
- Given graph G
- run MCMC to equilibrium
- then, for each sampled $\mathcal{D}$, draw a **resampled** graph G' from ensemble

A test: do resampled graphs look like original?

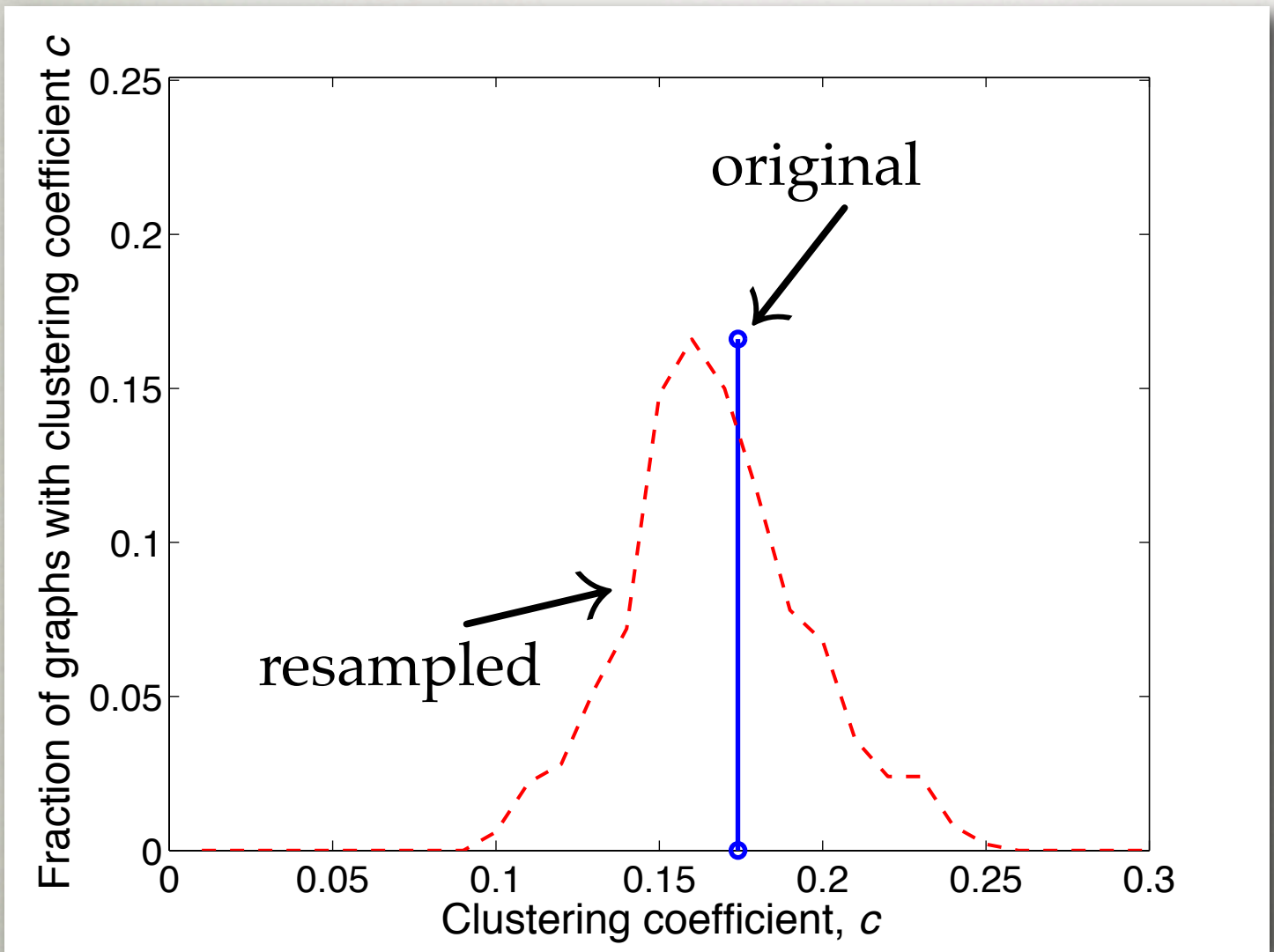

herbivore
plant
parasite

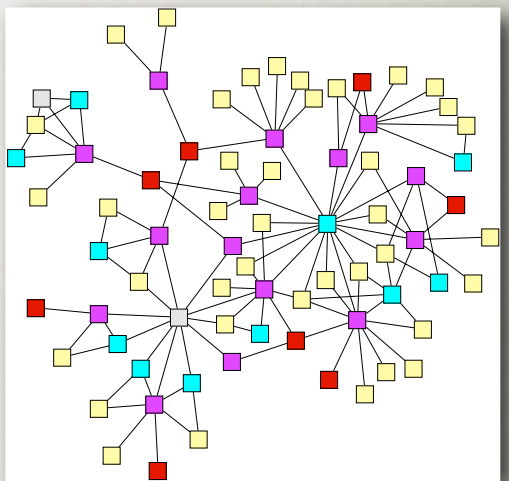
Degree Distribution



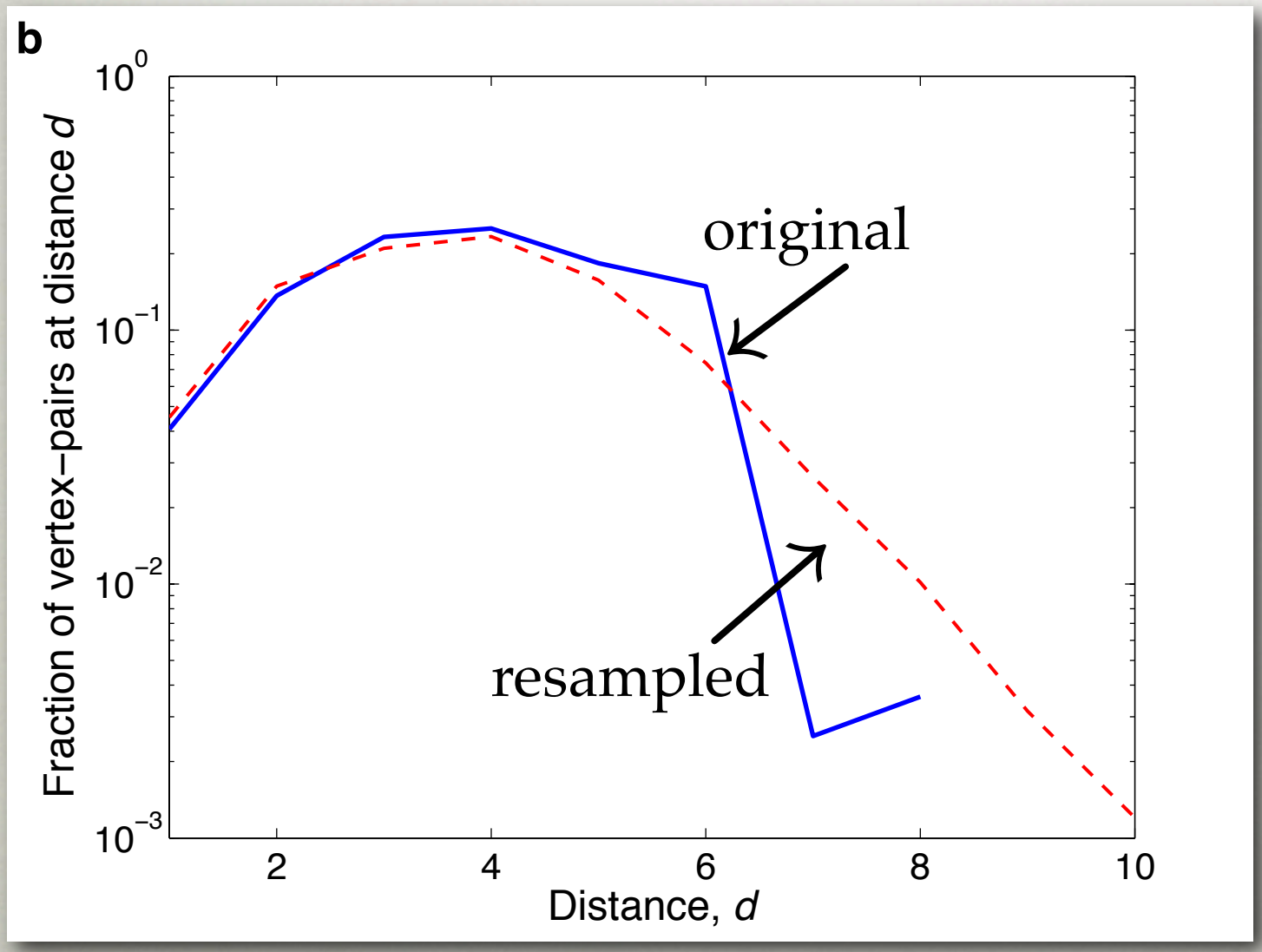


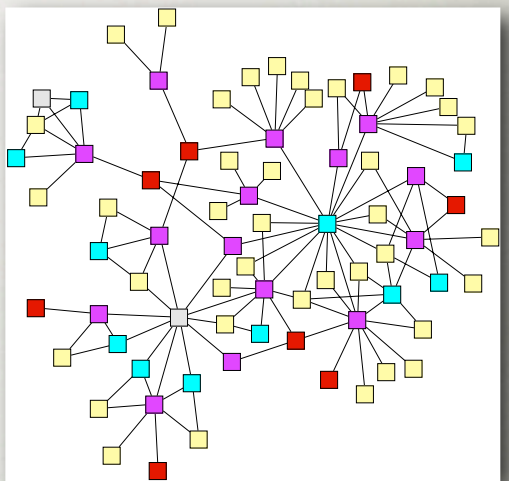
Clustering Coefficient





Distance Distribution





Missing Links

many networks partially known, noisy

- social nets, foodwebs, protein interactions, etc.

can hierarchies predict their **missing links**?

previous approaches

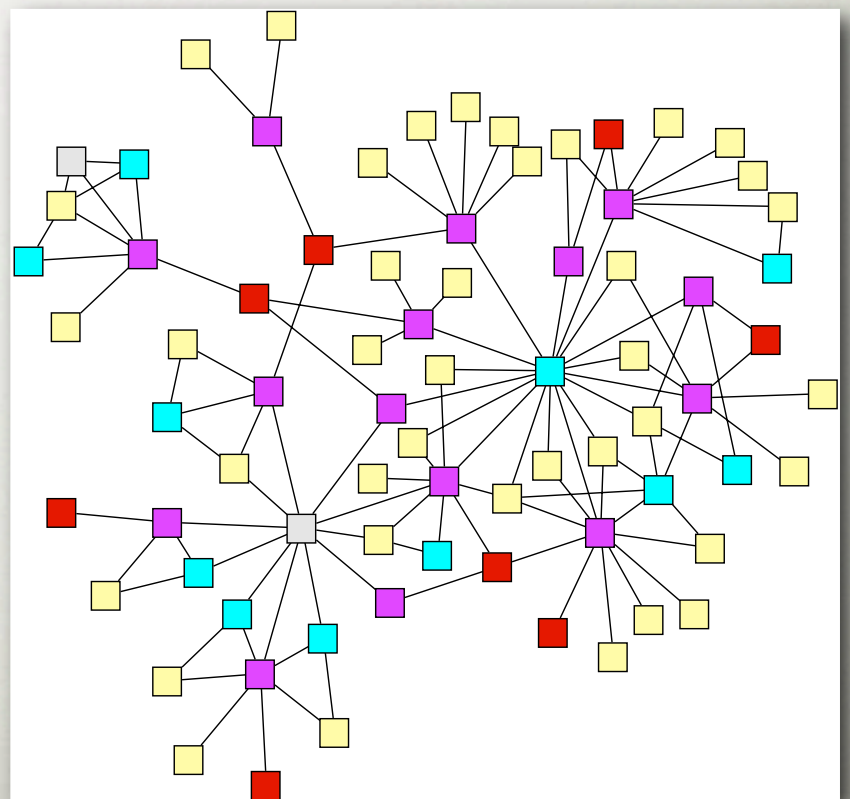
- Liben-Nowell & Kleinberg (2003)
- Goldberg & Roth (2003)
- Szilágyi et al. (2005)
- many more now

Accuracy is Hard

- remove k edges from G
- how easy to guess a missing link?

$$p_{\text{guess}} \approx \frac{k}{n^2 - m + k}$$
$$= O(n^{-2})$$

$$n = 75$$
$$m = 113$$
$$p_{\text{guess}} = k/(2662 + k)$$



An HRG Approach

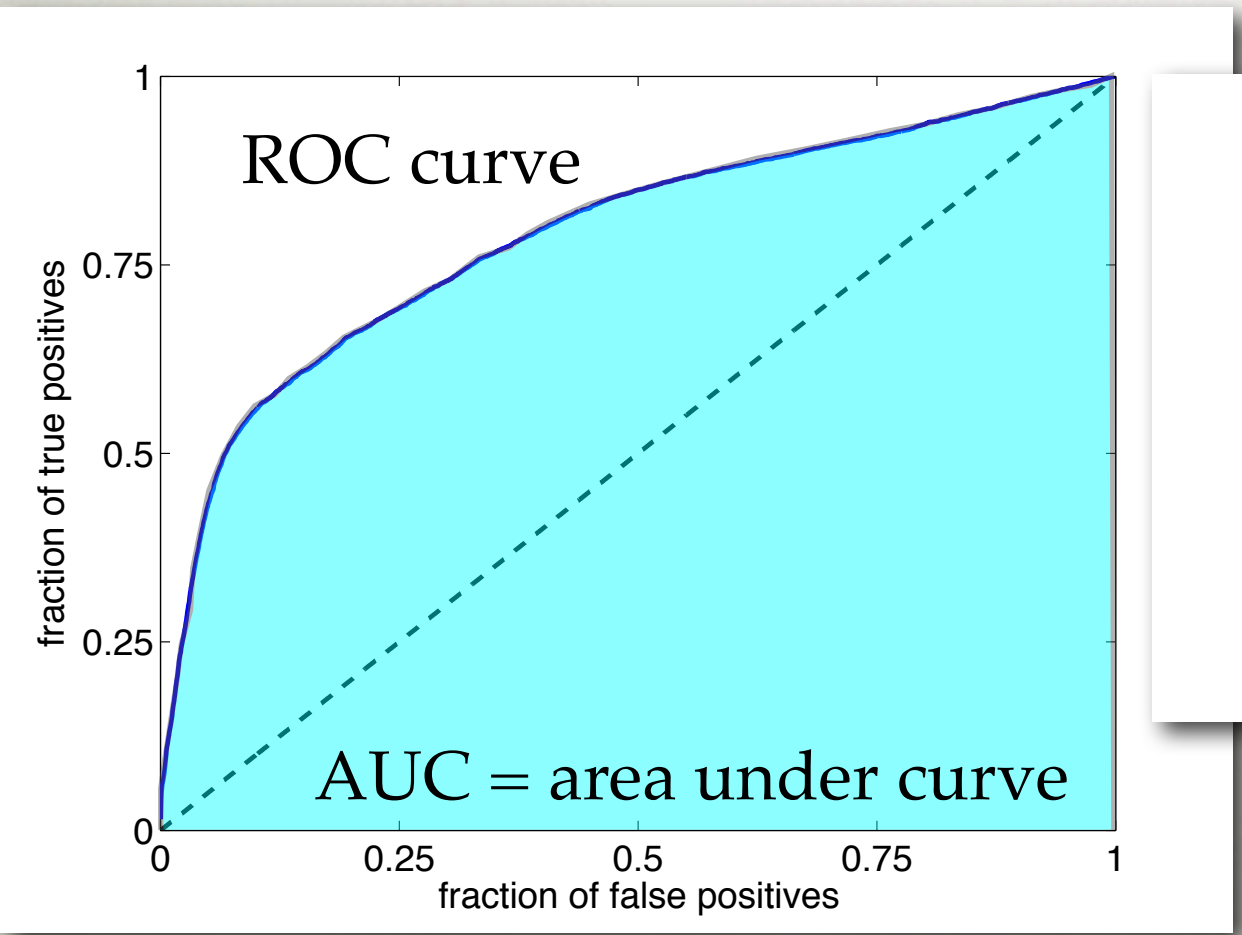
- Given incomplete graph G
- run MCMC to equilibrium
- then, over sampled $\mathcal{D}$, compute average $\langle p_r \rangle$ for links $(i, j) \notin G$
- predict links with high $\langle p_r \rangle$ values are missing

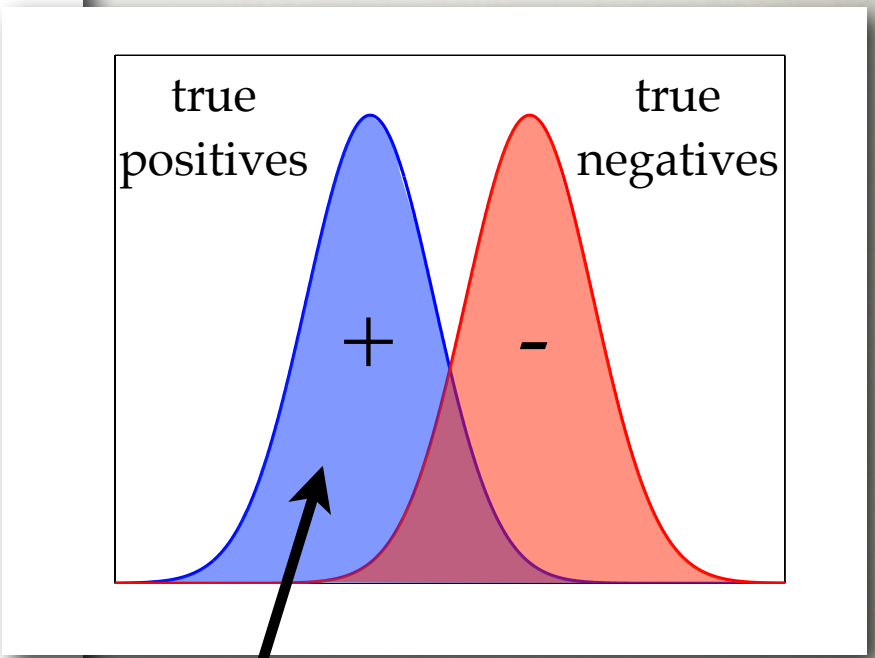
Test via leave-*k*-out cross-validation

perfect accuracy: AUC = 1

no better than chance: AUC = 1/2

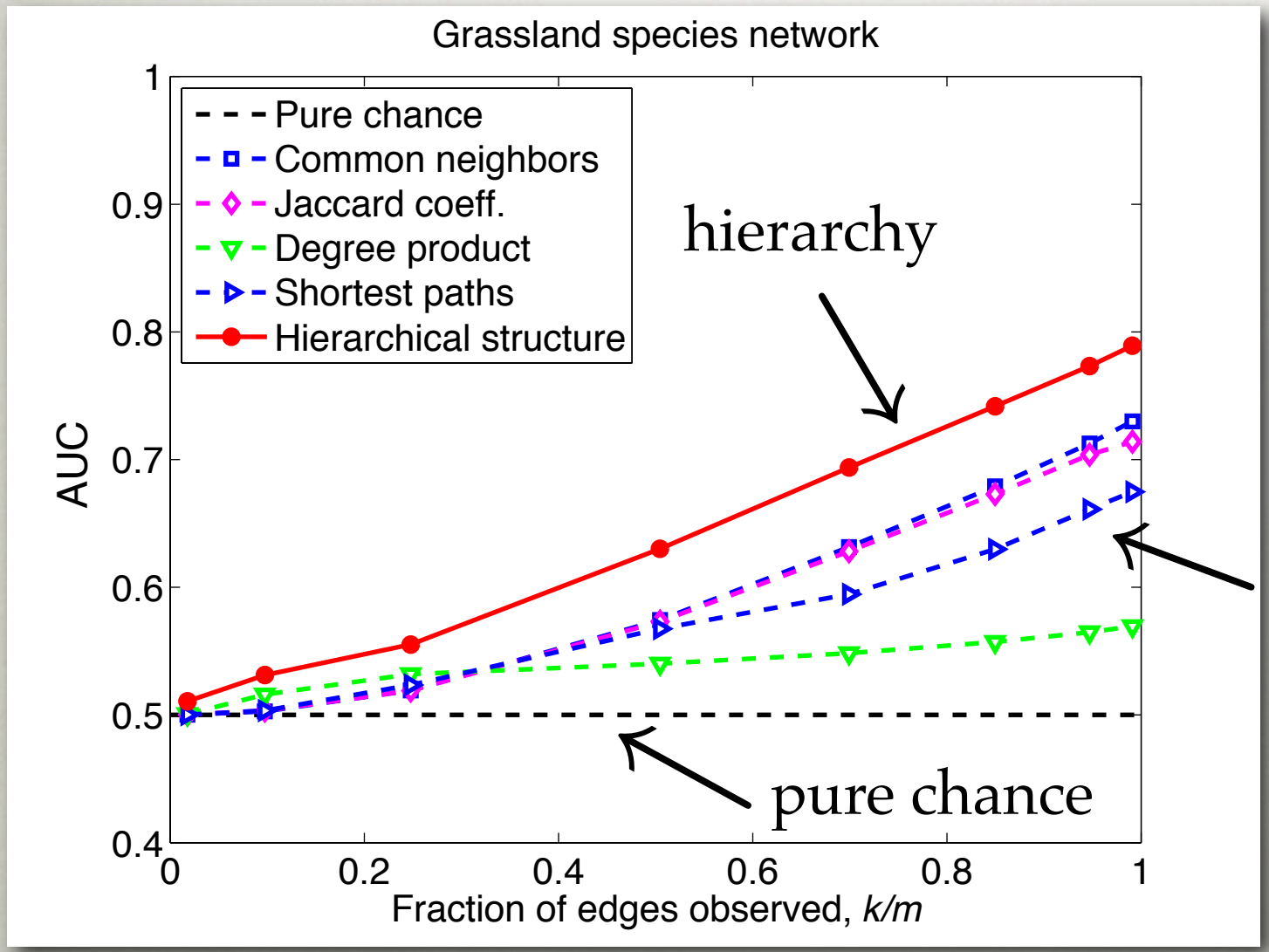
Scoring the Predictions

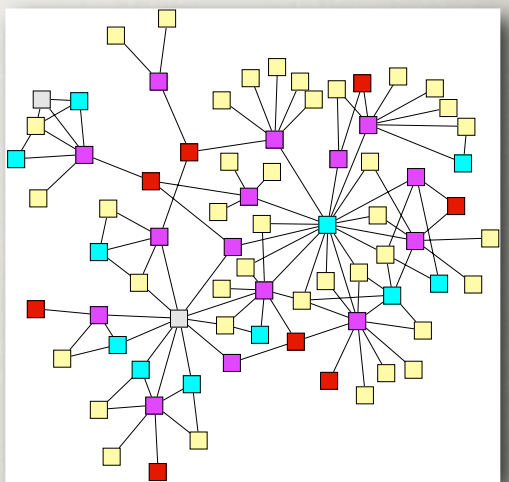




AUC =
Pr(distinguish + from -)

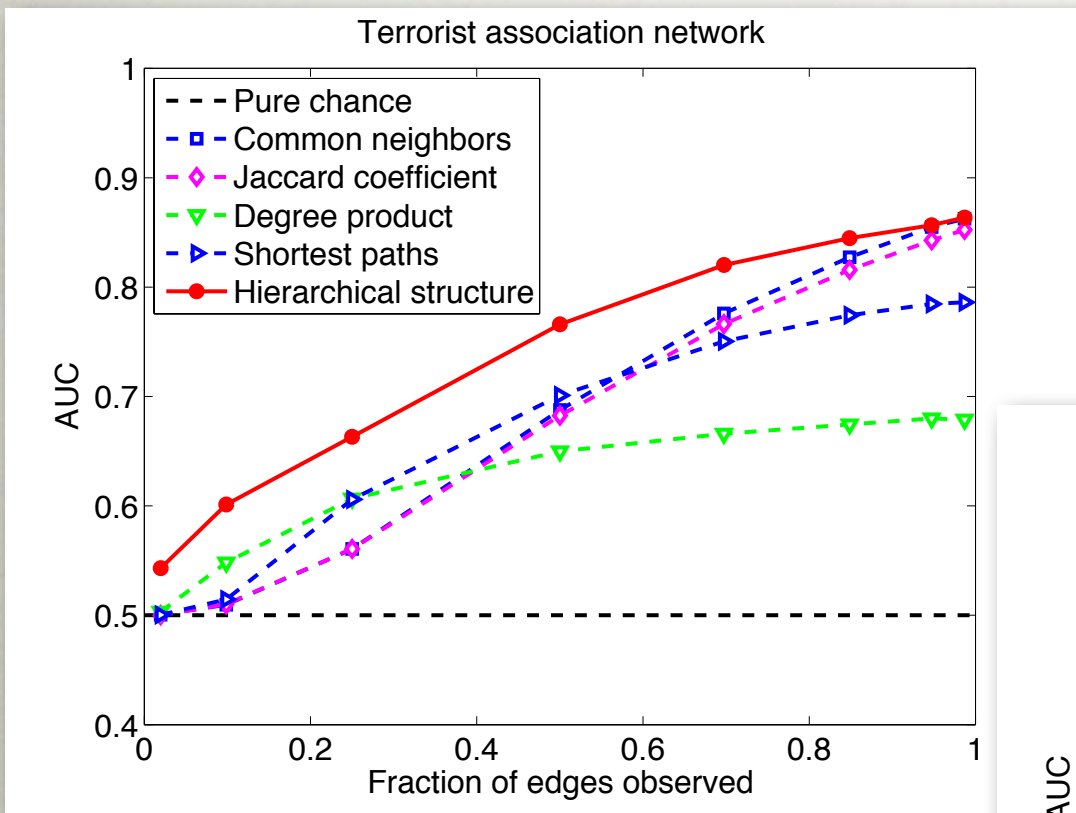
Performance 1

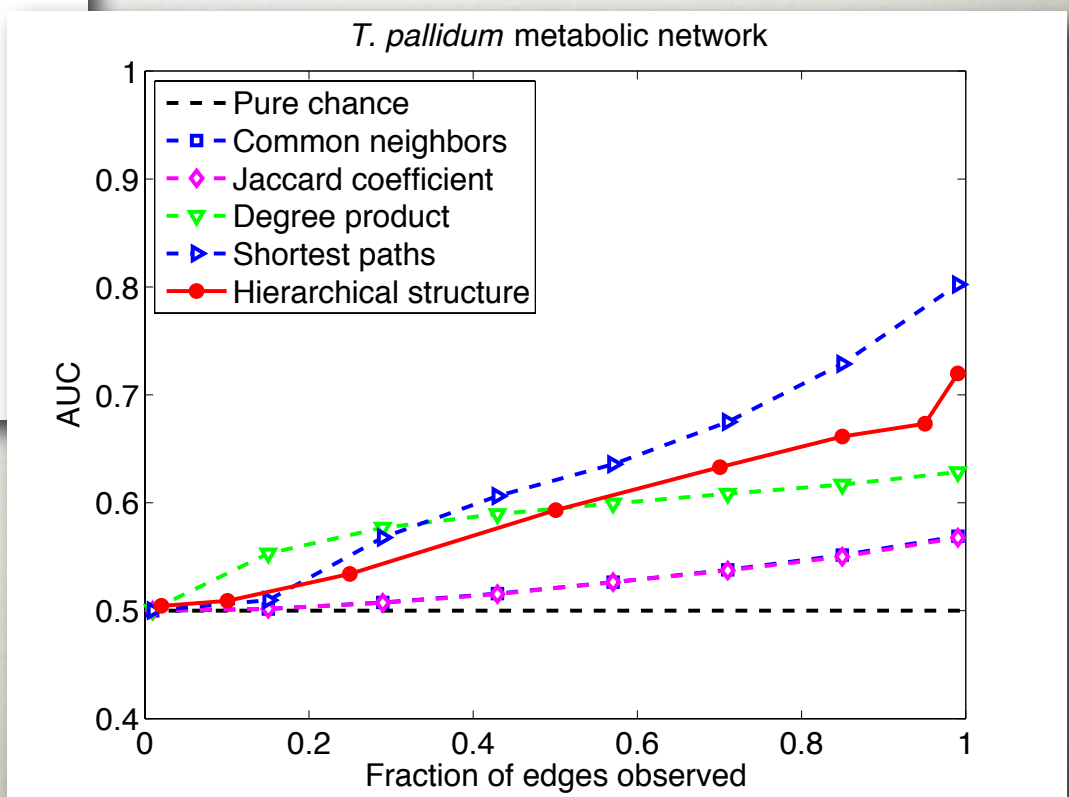




simple predictors

Performance 2





Some Final Thoughts

- what processes create these hierarchical structures?
- scaling up the running time from $O(n^2)$?
- active learning
- generalization to weighted, directed edges
- generalization to non-Poisson distributions

Fin