# Network Science

## #16 Other approaches to community detection

© 2020 T. Erseghe

MIME.

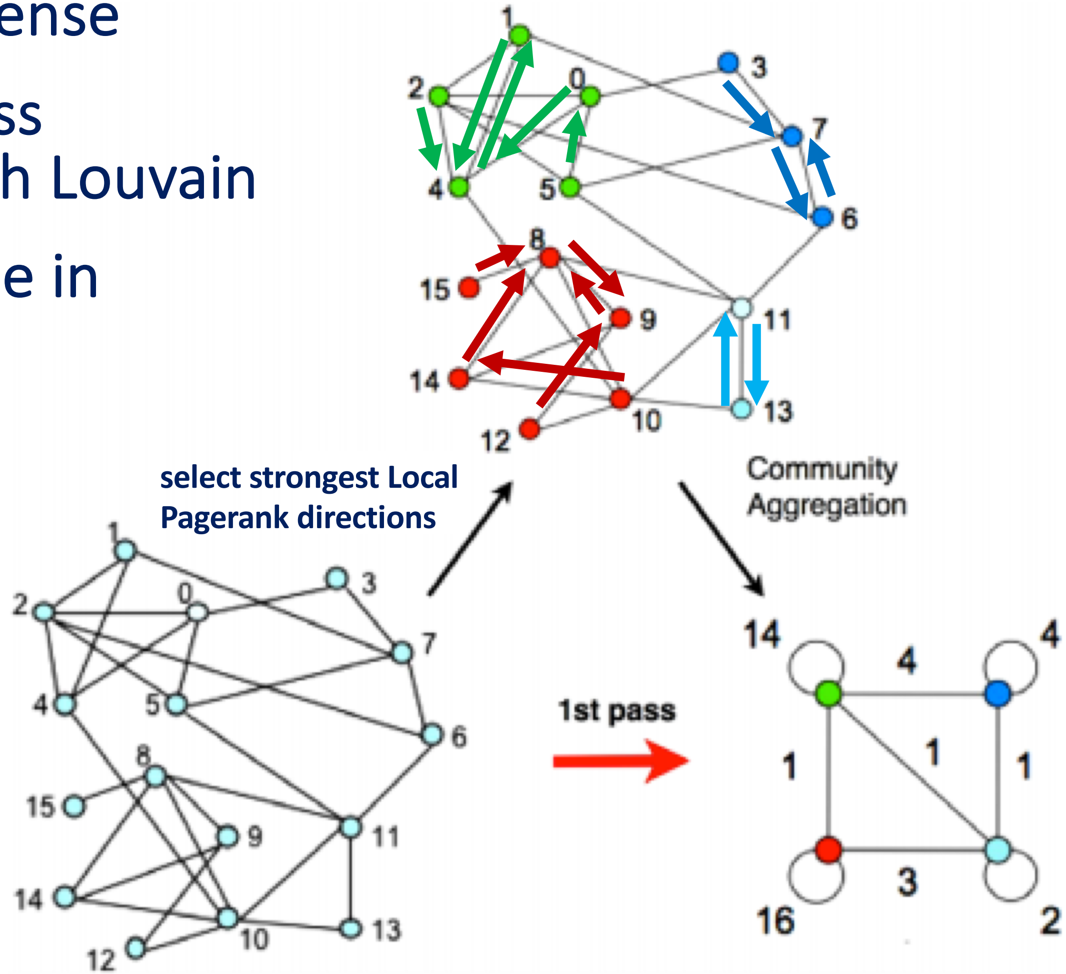
# Authority shift

Cho & MuLee (2010)
Authority-shift clustering: Hierarchical clustering by authority seeking on graphs
https://ieeexplore.ieee.org/abstract/document/5540081

# Authority shift

Idea

- ❑ Connect a node towards the strongest direction in the Local PageRank sense
- ❑ Iterate the process recursively as with Louvain
- ❑ Stop if no increase in modularity




MIME.

# Characteristics

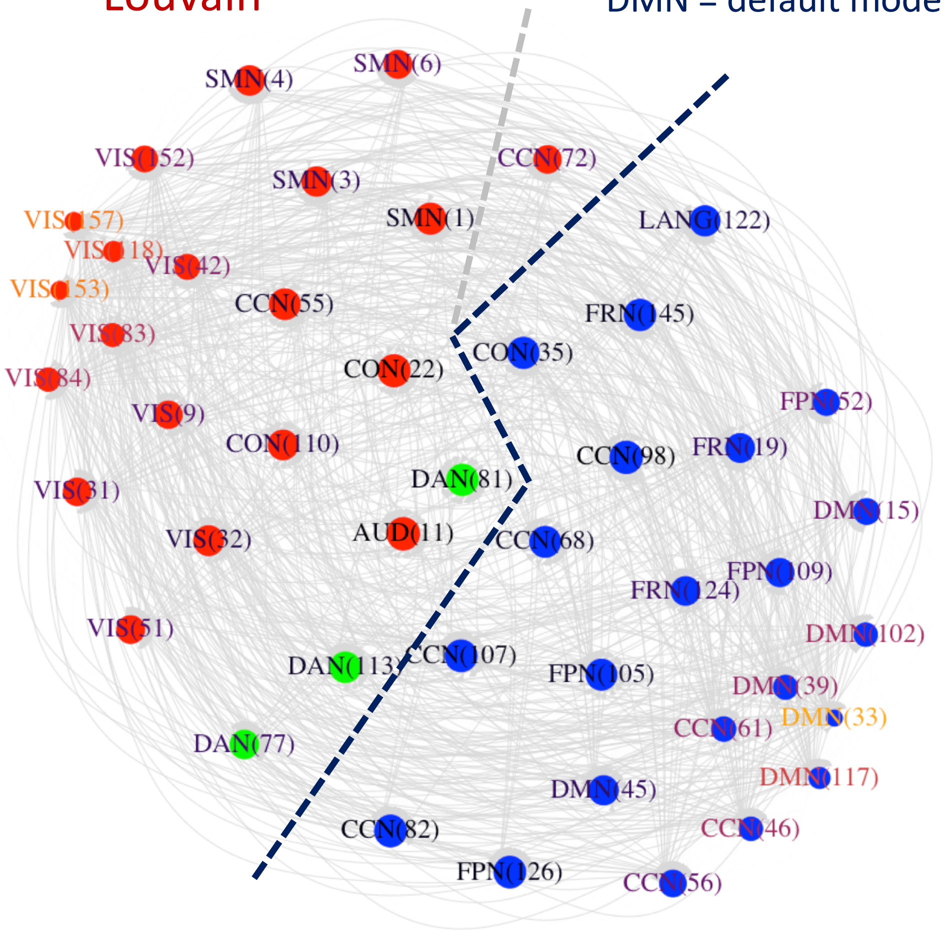
- Implements modularity optimization
- Scalable (low complexity)
- Effective
- Local PageRank can be efficiently calculated
- Not a greedy technique
- Easily extendable to networks with signs
- No parameters to be set

# Brain Network Example

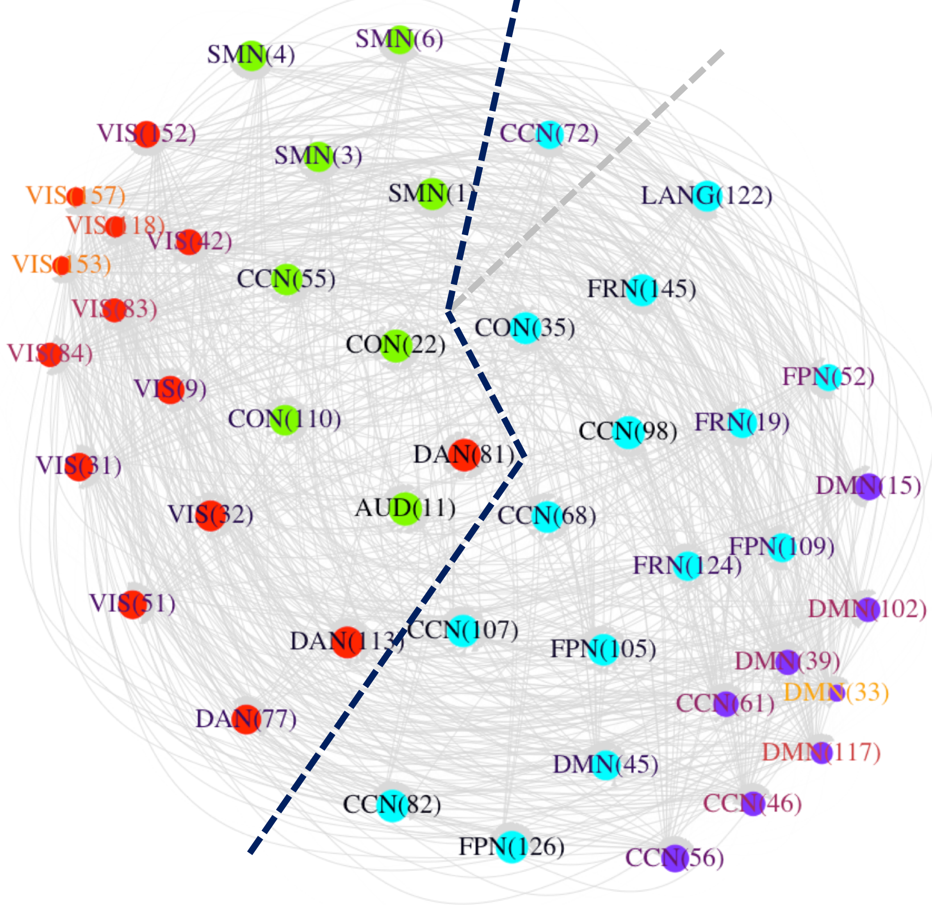
AUD = auditory
CCN = cognitive control
CON = cingulo-opercular
DAN = dorsal attention
DMN = default mode

FPN = fronto-parietal
FRN = feedback-related negativity
LANG = language
SMN = somato-motor
VIS = visual

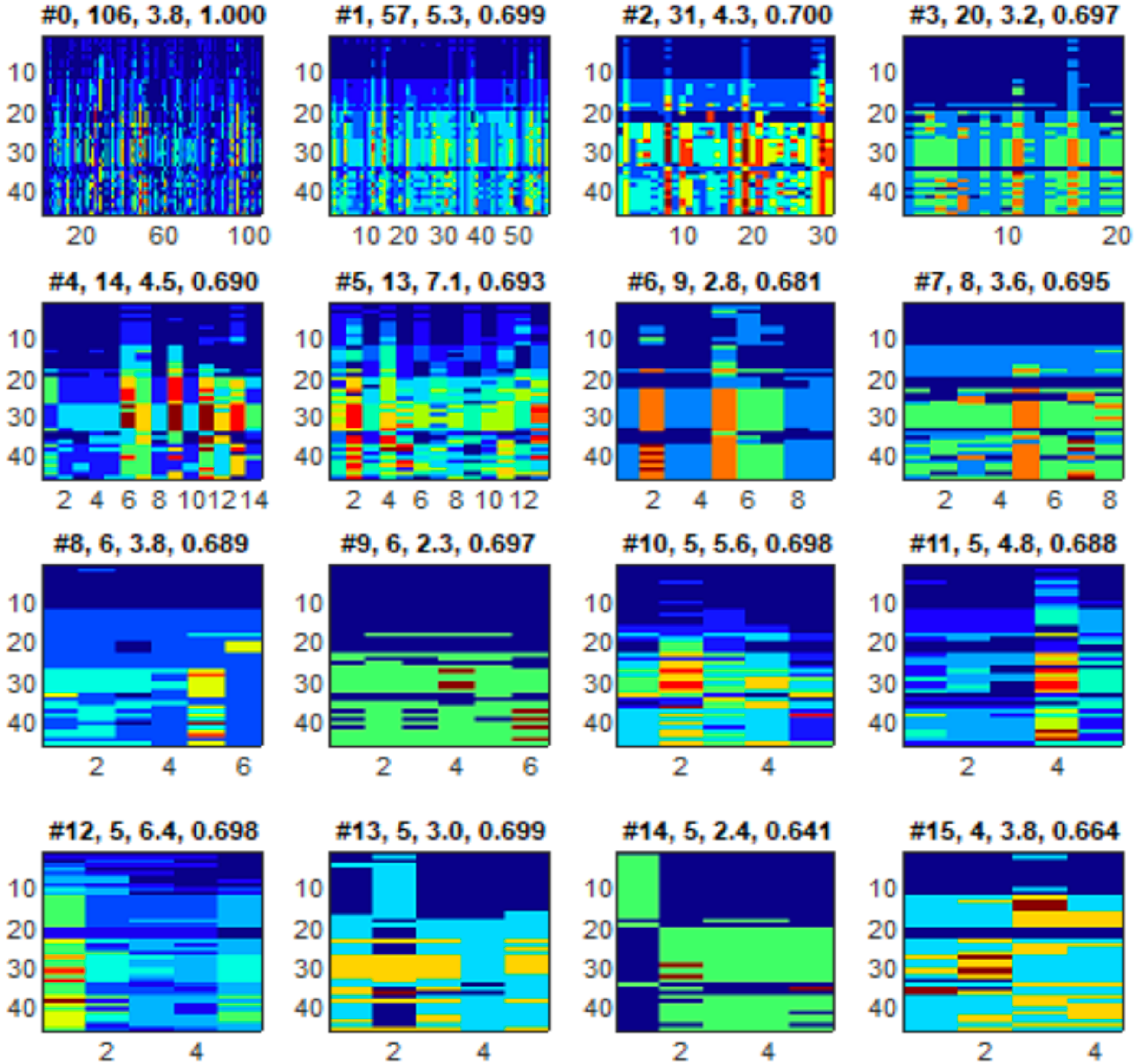
Louvain




Authority Shift

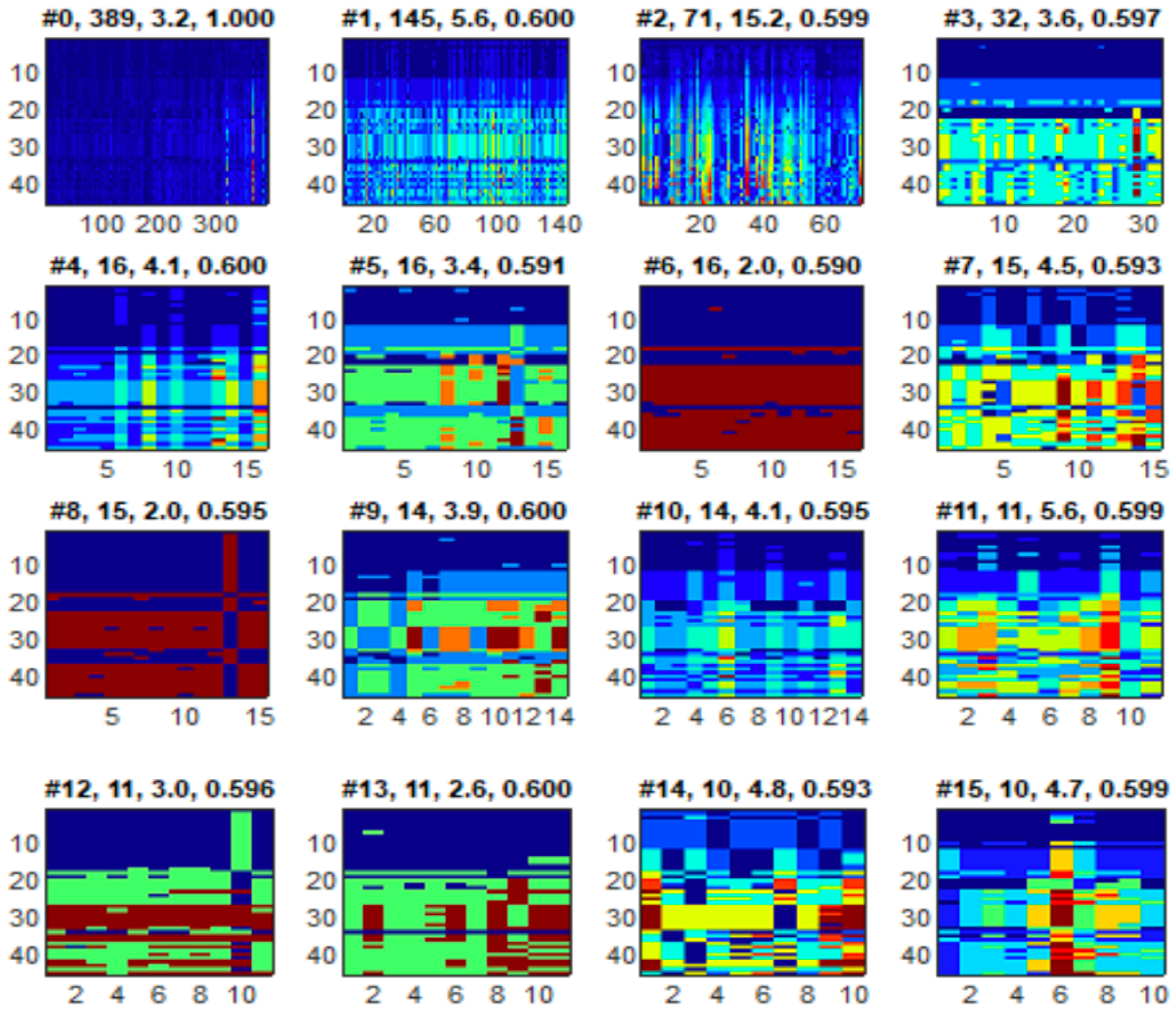


# Authority Shift – Subjects Clusters




MiME.

# Louvain – Subject Clusters



# Clique percolation

# Clique percolation

## Idea

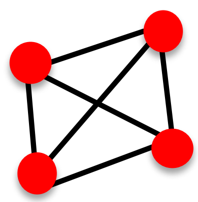
- Two nodes belong to the same community if they can be connected through adjacent *k* cliques

## *k* clique

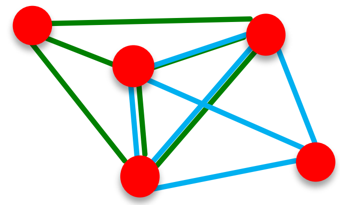
- Fully connected graph of *k* nodes

## Adjacent *k* cliques

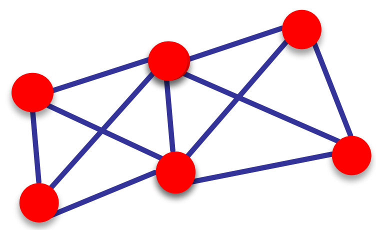
- Overlap in *k-1* nodes



4-clique



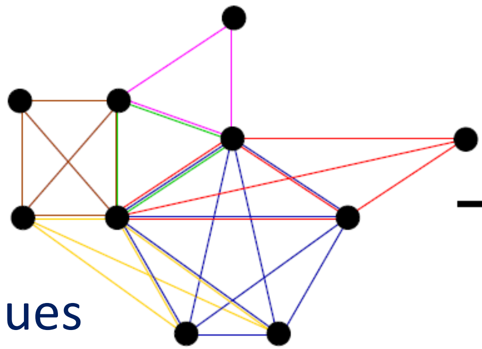
Adjacent 4-cliques



Non-adjacent 4-cliques

# Clique percolation

(1) identify cliques



**Graph**

Cliques

Overlap size

| | | | | | | |
|---|---|---|---|---|---|---|
| | 5 | 3 | 2 | 1 | 3 | 1 |
| | 3 | 4 | 2 | 1 | 1 | 1 |
| | 2 | 2 | 3 | 2 | 1 | 2 |
| | 1 | 1 | 2 | 3 | 0 | 1 |
| | 3 | 1 | 1 | 0 | 4 | 2 |
| | 1 | 1 | 2 | 1 | 2 | 4 |

**Clique overlap matrix**

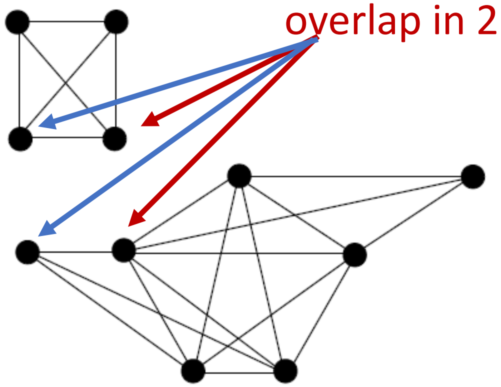
(2) Build a clique overlap matrix

k=4

| | | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 1 | 0 | 0 | 1 | 0 |
| | 1 | 1 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 |
| | 1 | 0 | 0 | 0 | 1 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 1 |

**Thresholded matrix at 3**

(3) Set a threshold (the one providing the richest community structure = most widely distributed cluster sizes)

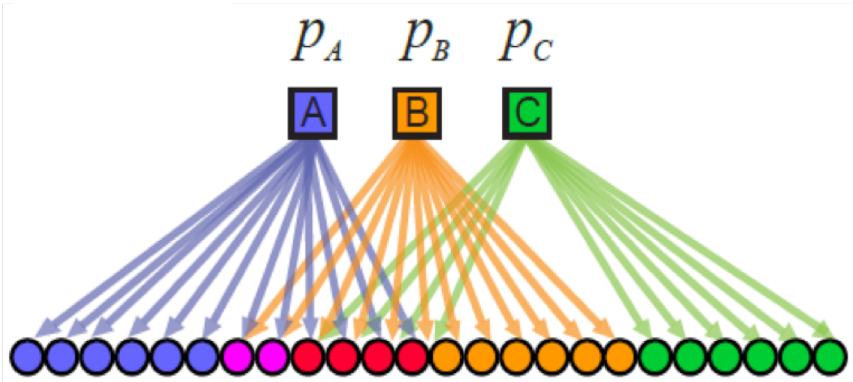


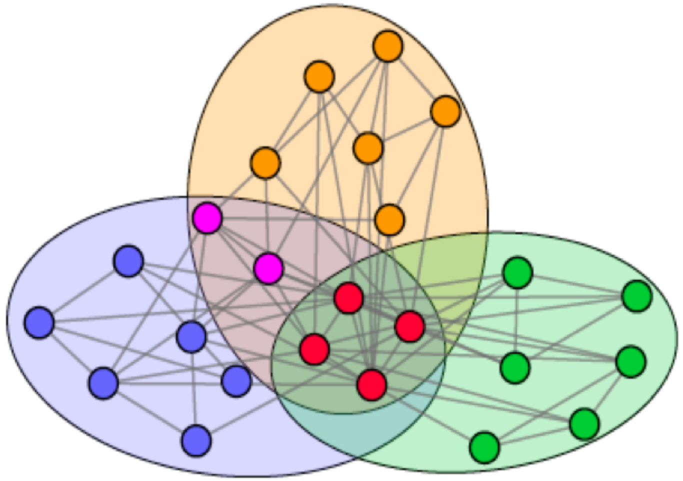

**Communities (connected components)**

# Big CLAM

Yang & Leskovec (2013)
Overlapping community detection at scale: a nonnegative matrix factorization approach
https://dl.acm.org/doi/pdf/10.1145/2433396.2433471?casa_token=ENfVUy8WEaUAAAAA:DDh024Jw5wchw69gYXm3BR3NRKDBjiDNz0pXRoJAWTKQD6HyT1iMRvc64WcclS7GLAAj30I-6Kiu

# BigCLAM



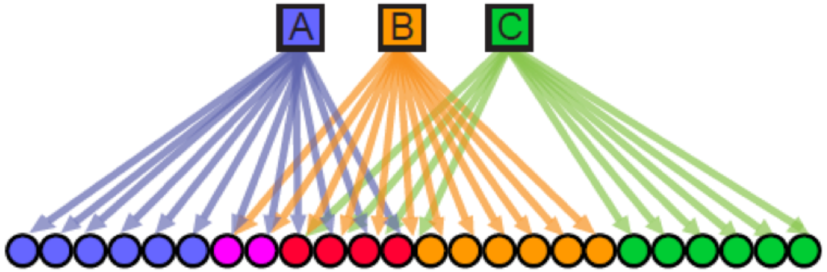



Affiliation graph model (AGM)

- Nodes $N$
- Communities $C$
- Membership matrix $\boldsymbol{M}$, binary $|N|\text{x}|C|$ matrix
- Probability $\boldsymbol{p} = [p_c]$ that a connection is active
- Probability that edge $(i,j)$ is not active $Q_{ij} = \prod_{c \in M_i \cap M_j} (1-p_c)$
- $\boxed{\boldsymbol{Q} = \exp(-\boldsymbol{M}\,\mathrm{diag}(\boldsymbol{q})\,\boldsymbol{M}^\top)}$ , $\boldsymbol{q} = -\log(\boldsymbol{1}-\boldsymbol{p}) > \boldsymbol{0}$

MIME

# BigCLAM




Target function $\boxed{f = \prod_{(i,j) \in \mathcal{E}} (1-Q_{ij}) \prod_{(i,j) \notin \mathcal{E}} Q_{ij}}$

active edges $\mathcal{E}$ is a known data

Recall that $\boldsymbol{Q} = \exp(-\boldsymbol{M}\ \mathrm{diag}(\boldsymbol{q})\ \boldsymbol{M}^{\mathrm{T}})$

Want to maximize $f$ w.r.t. $\boldsymbol{M}$ binary and $\boldsymbol{q} > 0$

- This is a maximum likelihood estimator (best fit in probability)
- Needs to know in advance the # of communities $|C|$
- Very difficult to solve exactly – NP-complex

MIME.

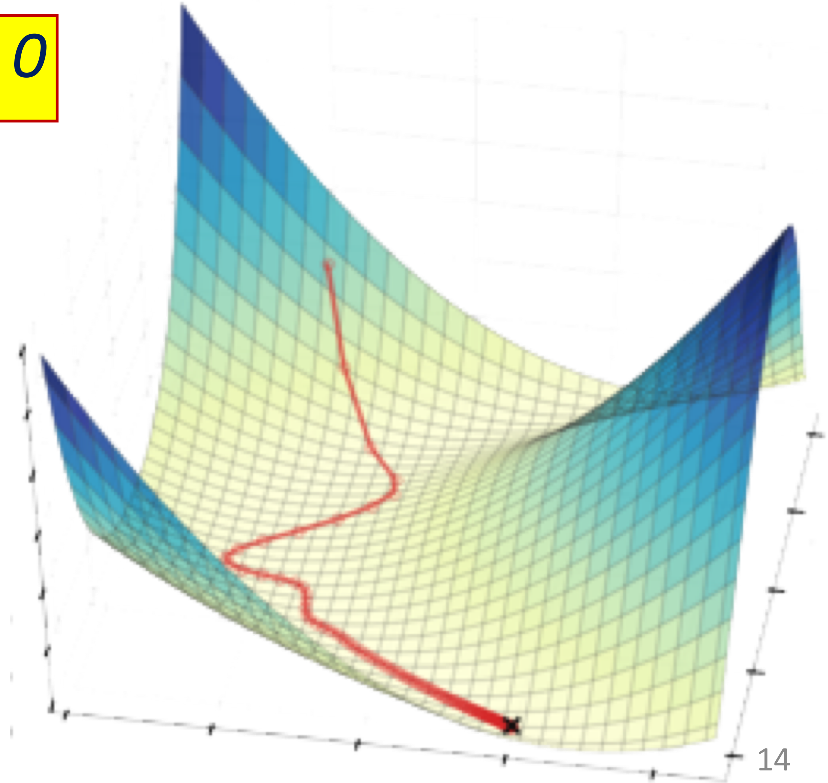
# BigCLAM

Idea: give mebership $\boldsymbol{M}$ a strength = relaxation !!!

Target function $f = \prod_{(i,j) \in \mathcal{E}} (1-Q_{ij}) \prod_{(i,j) \notin \mathcal{E}} Q_{ij}$

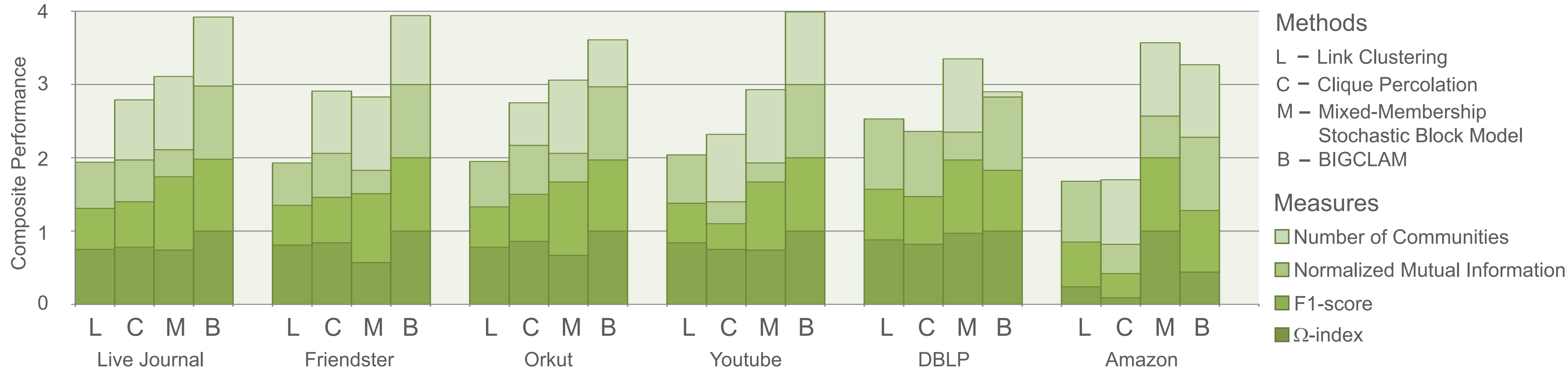
s. to $\boxed{\boldsymbol{Q} = \exp(-\boldsymbol{M}\,\boldsymbol{M}^{\mathrm{T}})}$ and $\boxed{\boldsymbol{M} > 0}$

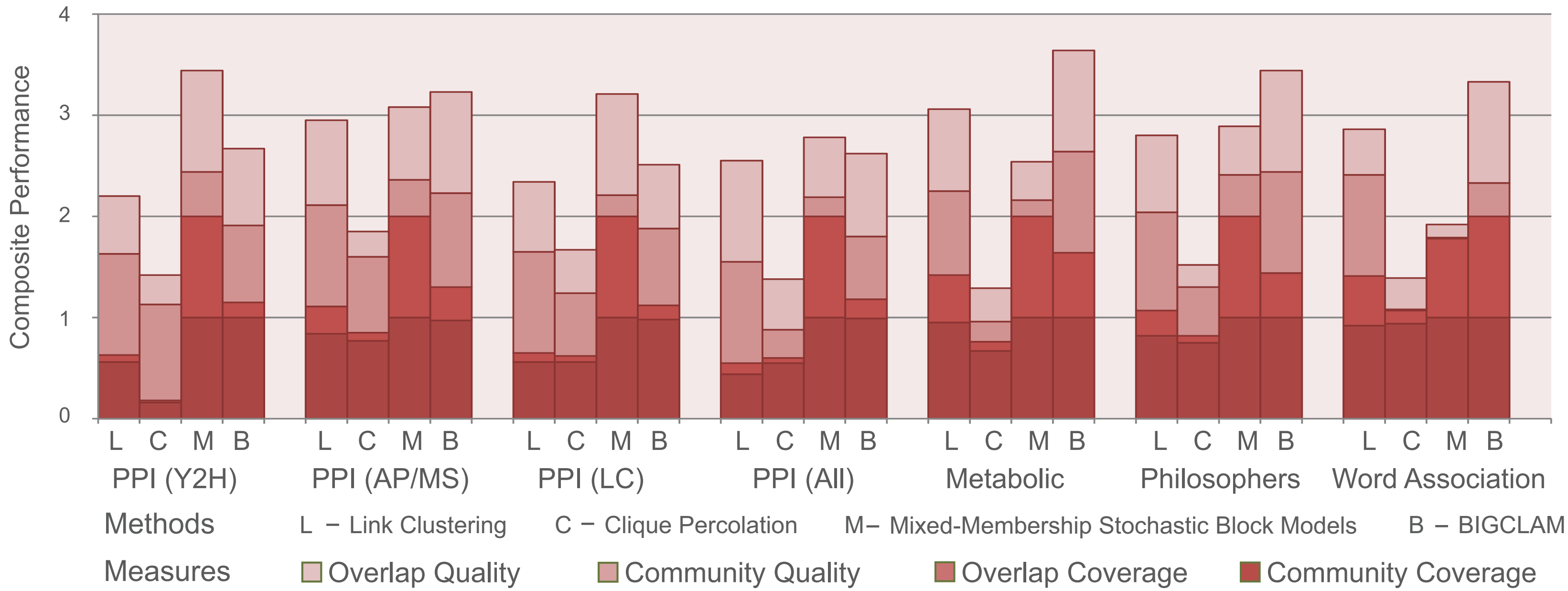
Can be solved using
gradient descent methods



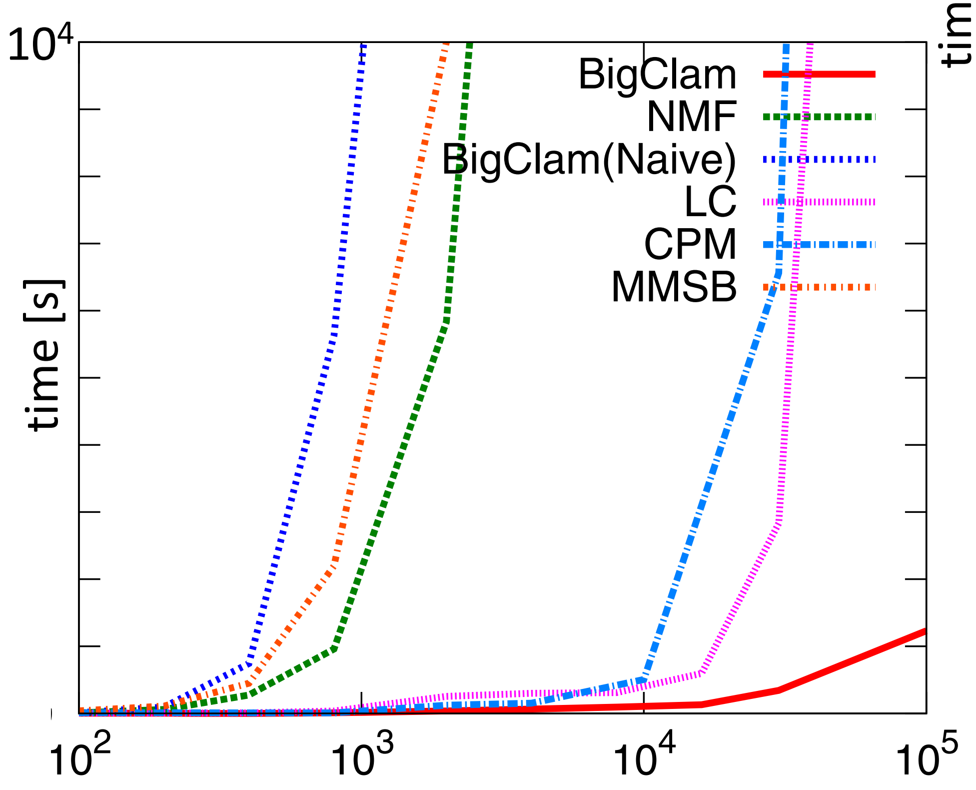
MIME.

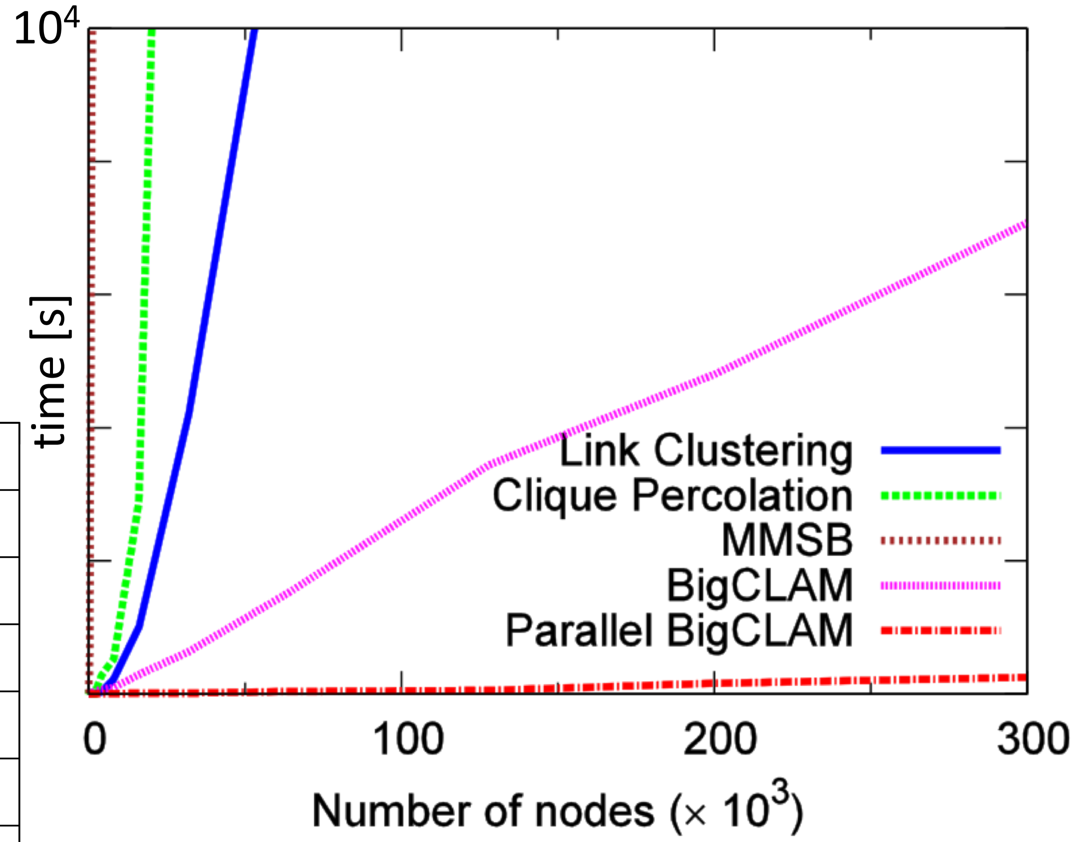
# Performance



# Performance



# Questions ?