GRAPH PROBLEMS FROM COMPLEX NETWORKS TEAM

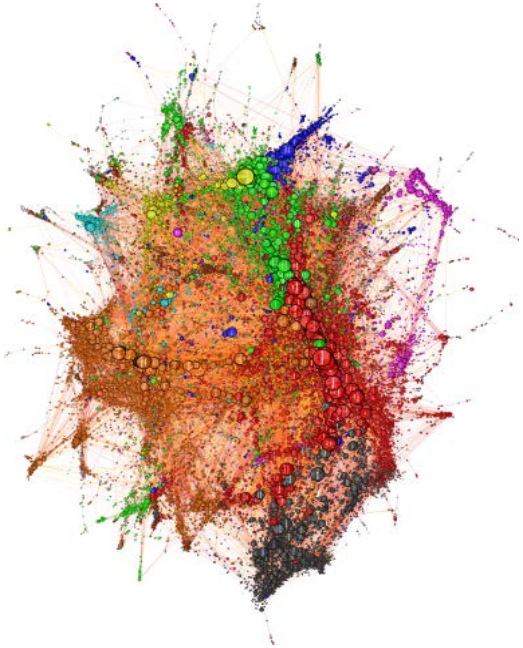
JEAN-LOUP GUILLAUME

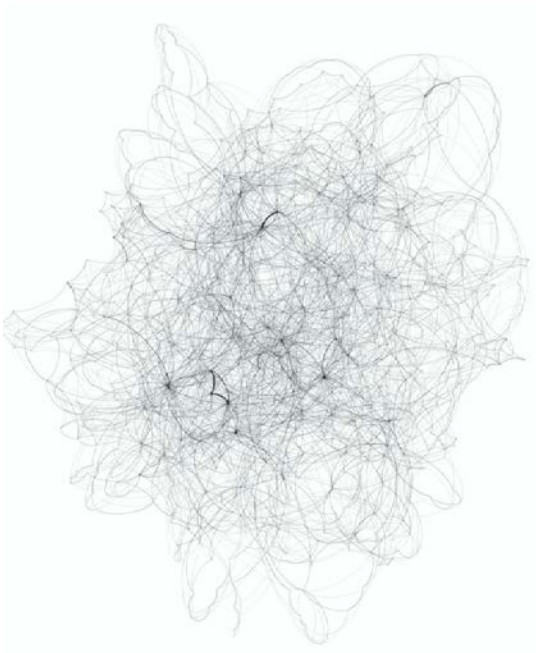
17-18/03/2014

JEAN-LOUP.GUILLAUME@LIP6.FR
COMPLEX NETWORKS TEAM
LIP6 - CNRS - UNIVERSITÉ PIERRE ET MARIE CURIE

Complex networks

- Relational data modeled using graphs:
  - Computer science: web, the Internet, email, P2P, …
  - Social sciences: friendships, collaborations, ...
  - Biology: neurons, proteins interactions, food chain, ...
  - Linguistics, transportation, ...





Need for specific algorithms?

- Size:
  - Internet = Millions of machines
    - More to come with the internet of objets
  - Facebook = over 1 billion users with 130 friends in average
  - Web = Google accounts for 10^15 distinct URLs

- It is not trivial to:
  - Store the networks in main memory
  - Do some computation

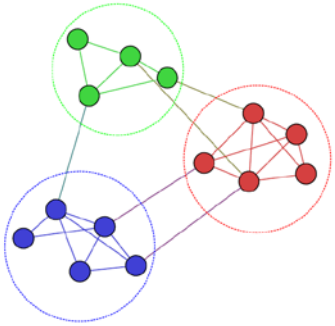
- Another definition of hard problems:
  - Anything over (and including) n^2

Examples

- Diameter of a graph:
  - Highest shortest distance between any two nodes
  - Theoretical complexity: O(n.m)
  - Approximation (without proof) using upper and lower bound in O(m)

- Count the number of triangles of a graph:
  - Naive approach $O(n^3)$
  - $O(m.n^{1/a})$ if the degree distribution is a power law with exponent a

- Community detection:
  - NP-hard in general (using most classical definitions)
  - Can be computed in “linear time” on real networks



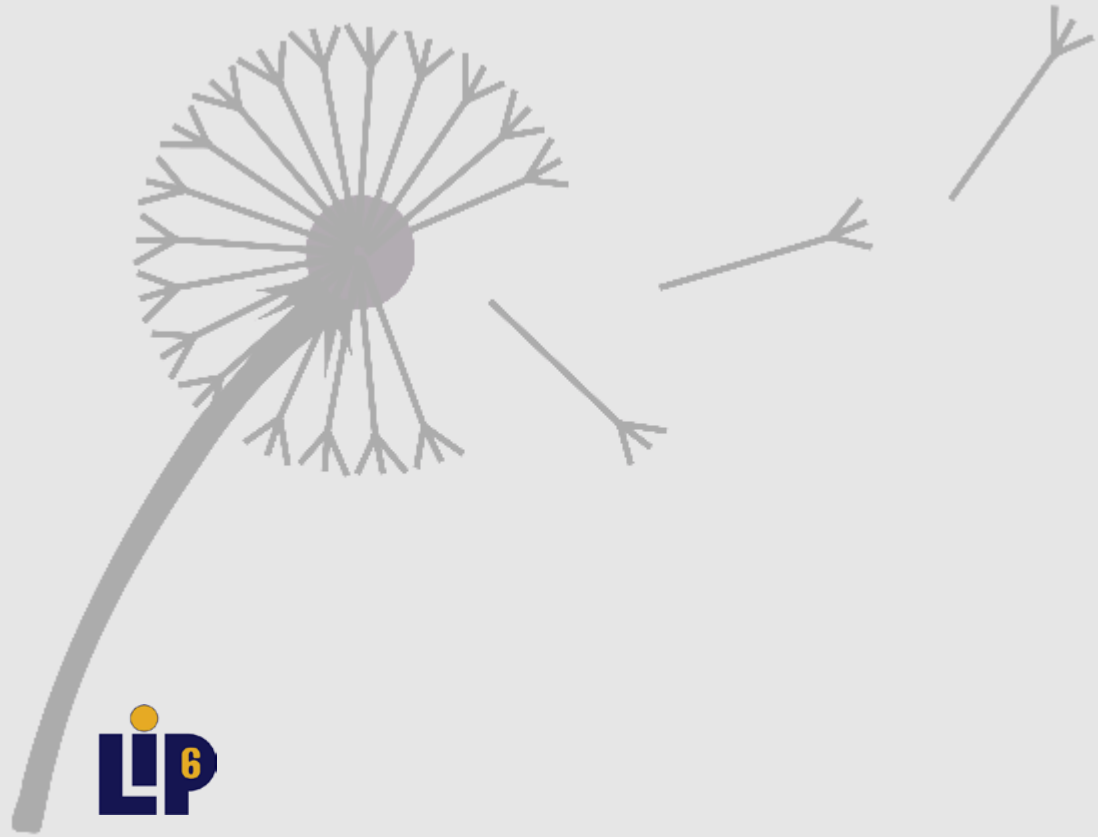
- How to take into account properties of complex networks?

Common properties/questions

- Most complex networks share many topological properties:
  - Low average distance / small world effect
  - Heterogeneous degrees / scale free networks
  - Clustering / variation of density and communities
  - Frequent motifs / triangles or more complex subgraphs

- Many similar studies on these networks:
  - Measurements + metrology
  - Description, modeling, simulation
  - Diffusion of information
  - Efficient algorithms design
  - …

COMMUNITY DETECTION

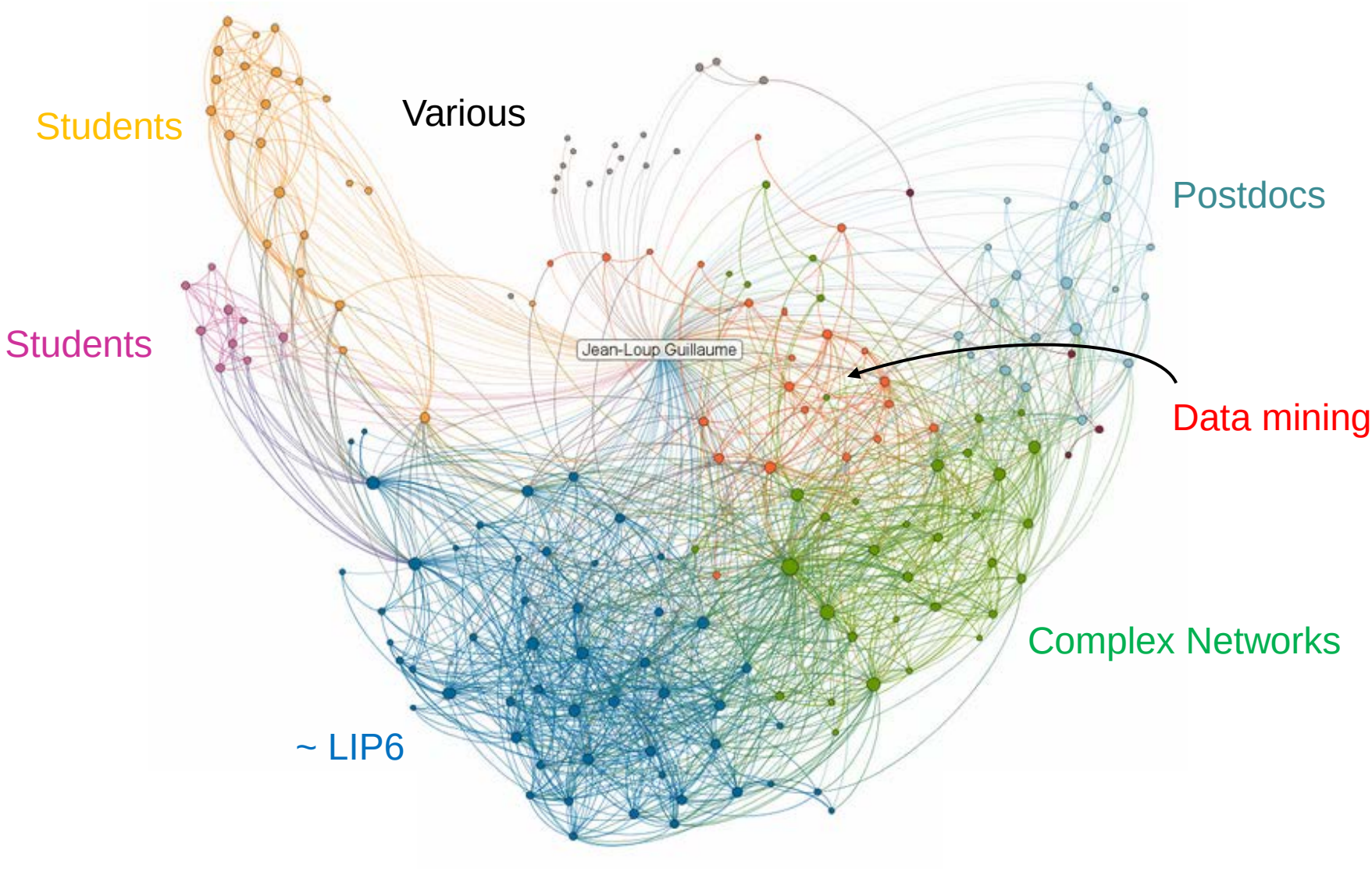


LIP6

UPMC SORBONNE UNIVERSITÉS

cnrs dépasser les frontières

Linkedin/inMaps - 03/2014



Modularity definition

- More links than expected in each group

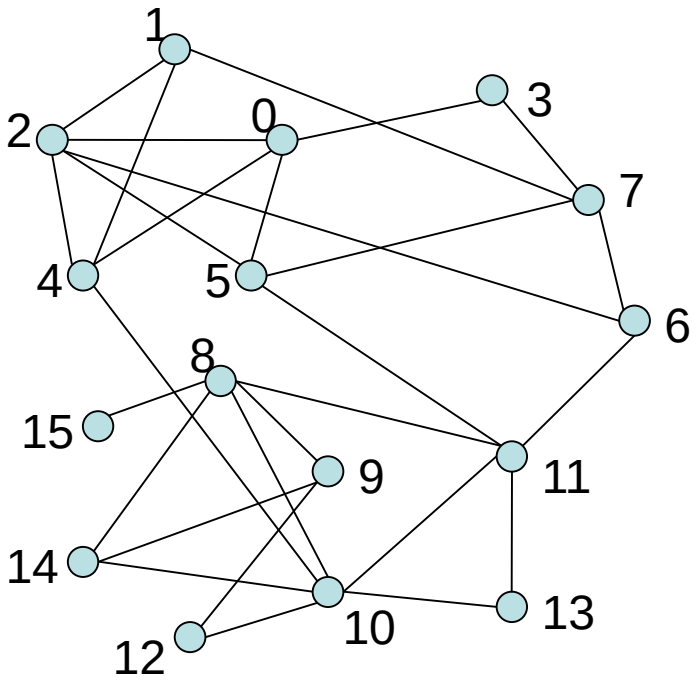
$$Q = \frac{1}{L}\sum_{s=1}^{m}\left(l_s - \frac{d_s^2}{4L}\right) = \sum_{s=1}^{m}\left[\frac{l_s}{L} - \left(\frac{d_s}{2L}\right)^2\right]$$

- ls : number of links within s
- L : total number of links

- Q: are groups more dense than expected?
  - Can tell whether a graph is modular or not.
  - Can also compare algorithms efficiency.

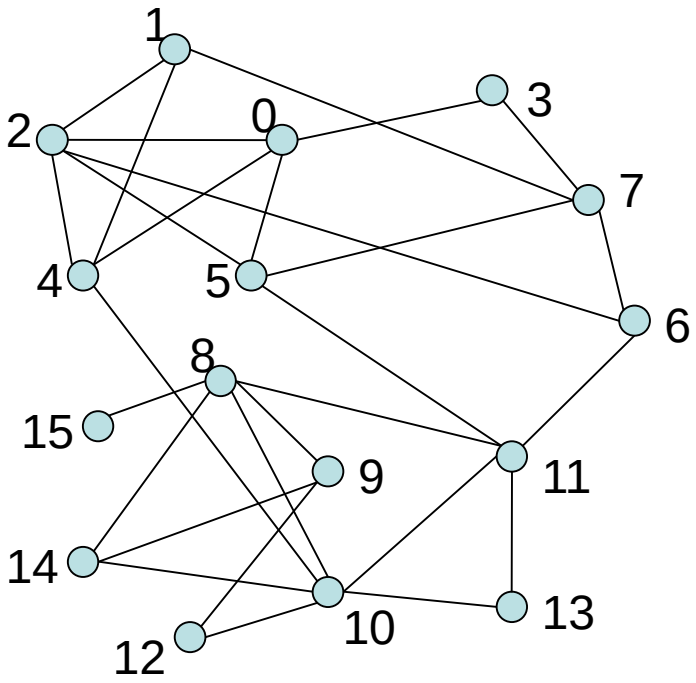
One example

- Graph with 16 nodes:
  - How many partitions?
  - How many connected partitions?
  - How many optimal partition (using modularity)?

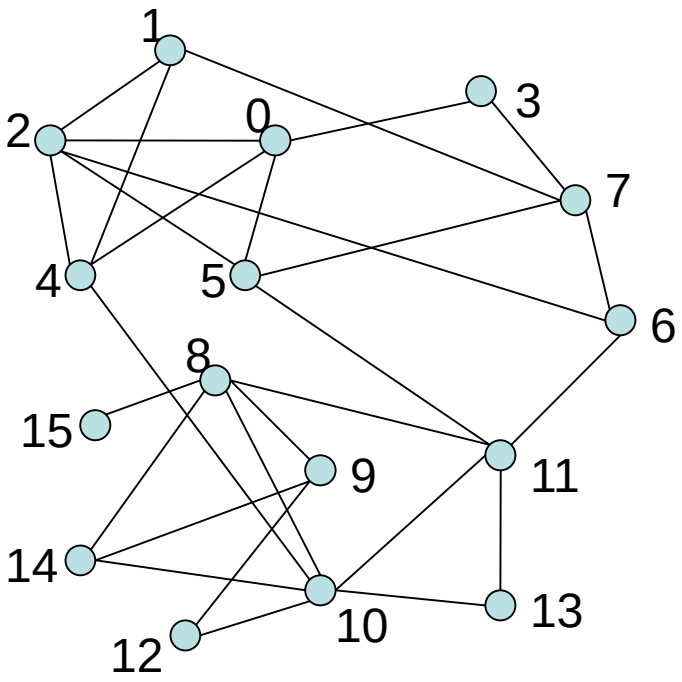


One example

- Graph with 16 nodes:
  - How many partitions? Around 10 billions (Bell number)
  - How many connected partitions? 44484
  - How many optimal partition (using modularity)? 1

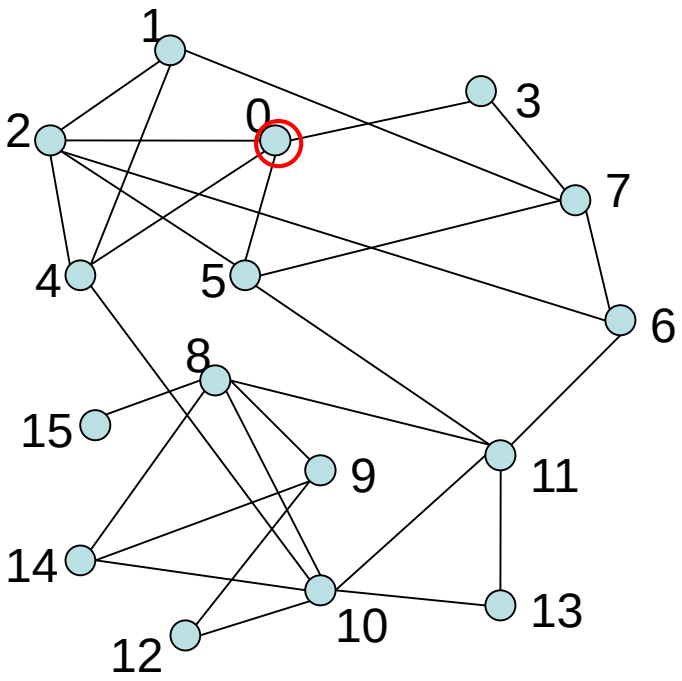


Louvain method – an example

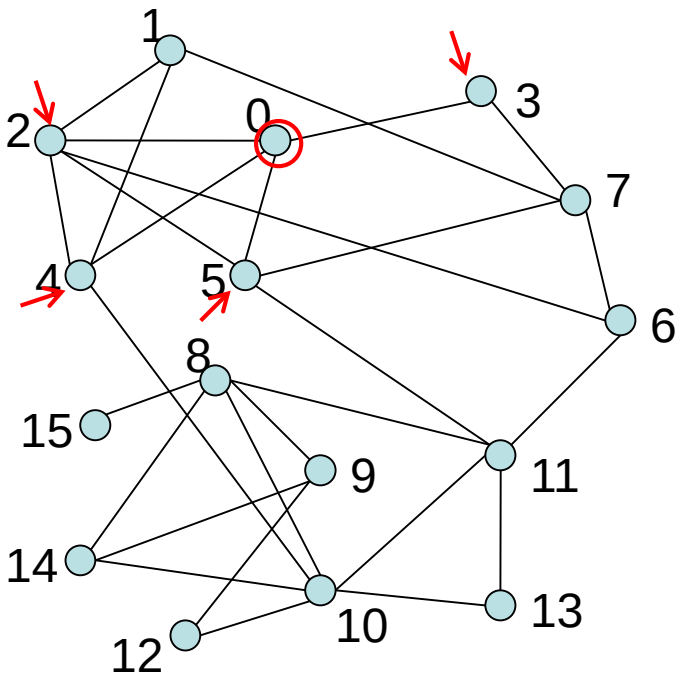


Initially: isolated nodes

Louvain method – an example

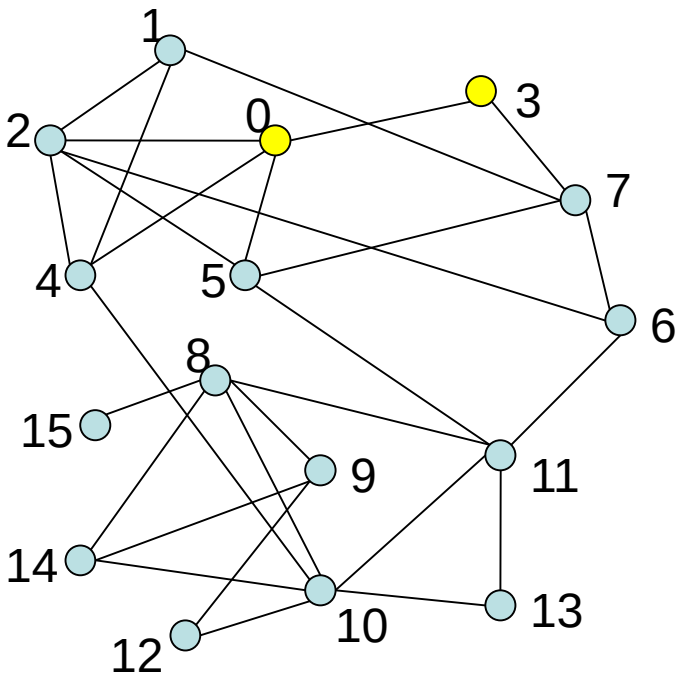


Louvain method – an example



Only neighbors are considered
Modularity gain is computed for each

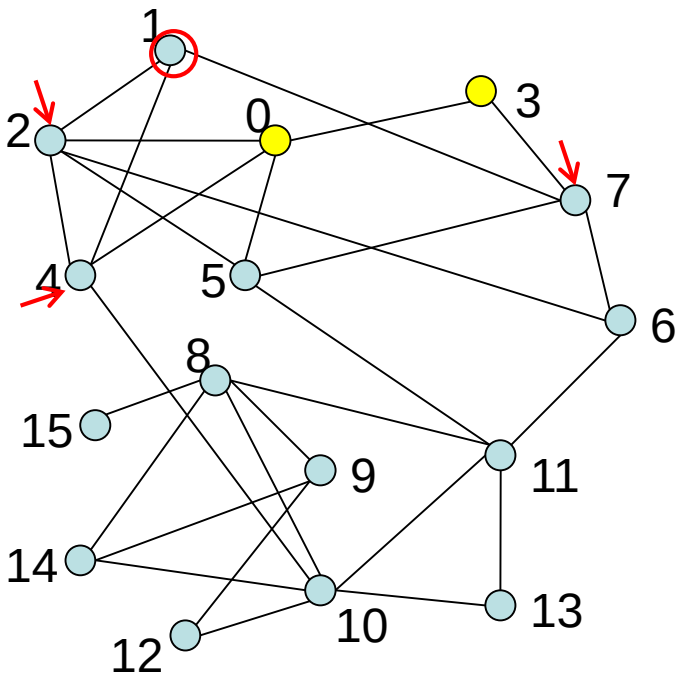
Louvain method – an example



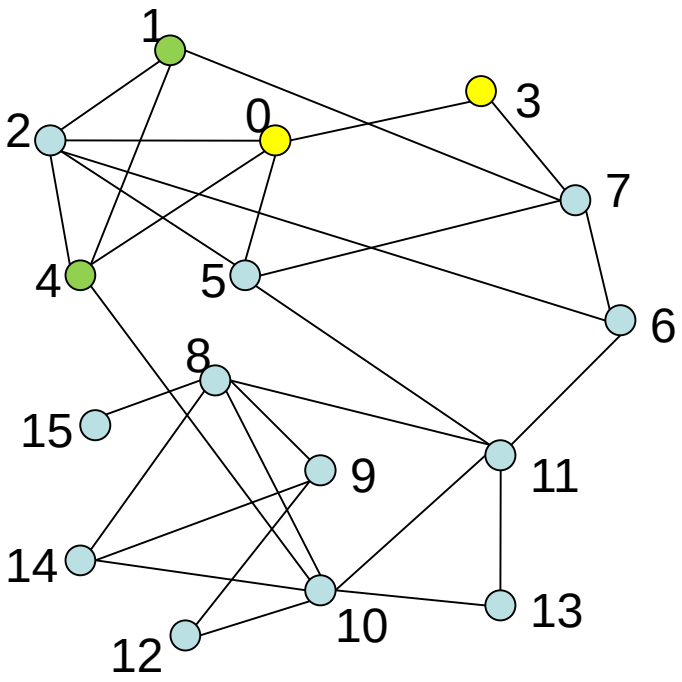
Pass 1 – Iteration 1
insert 0 in c[3]

Louvain method – an example

Pass 1 – Iteration 1
insert 0 in c[3]

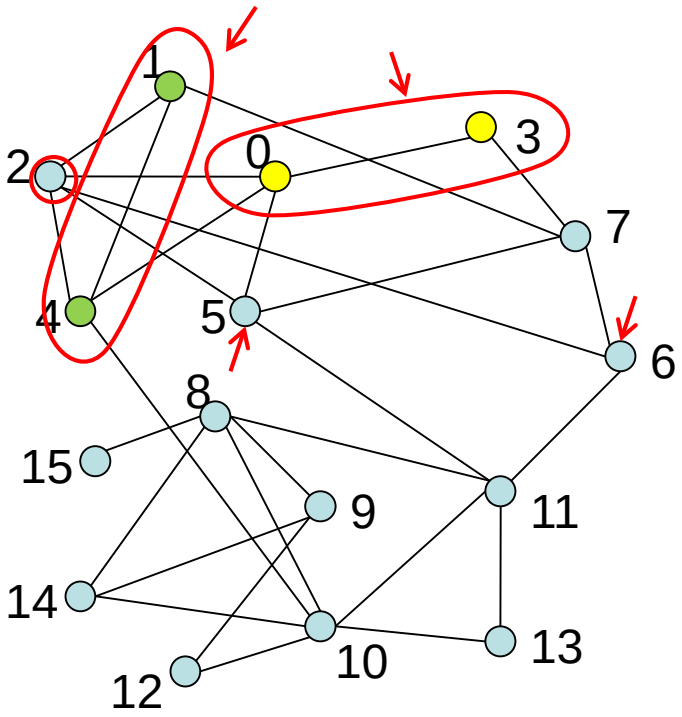


Louvain method – an example



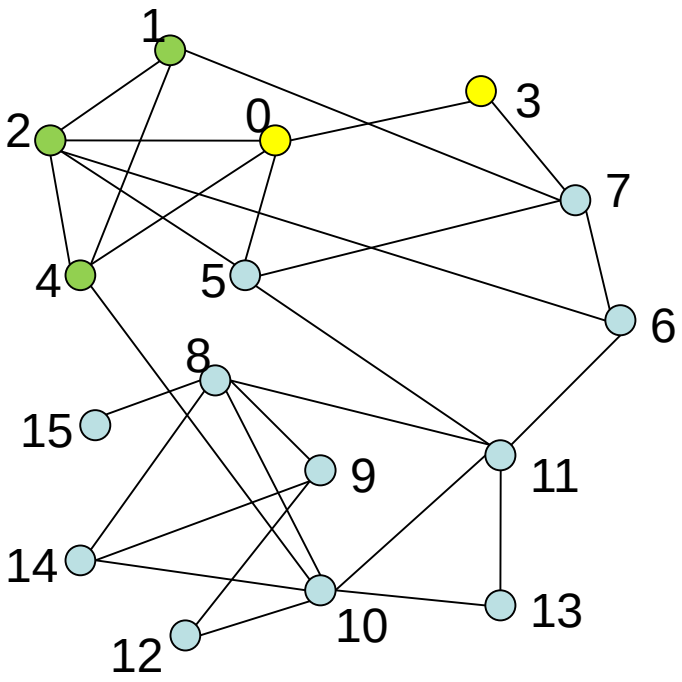
Pass 1 – Iteration 1
insert 0 in c[3]
insert 1 in c[4]

Louvain method – an example



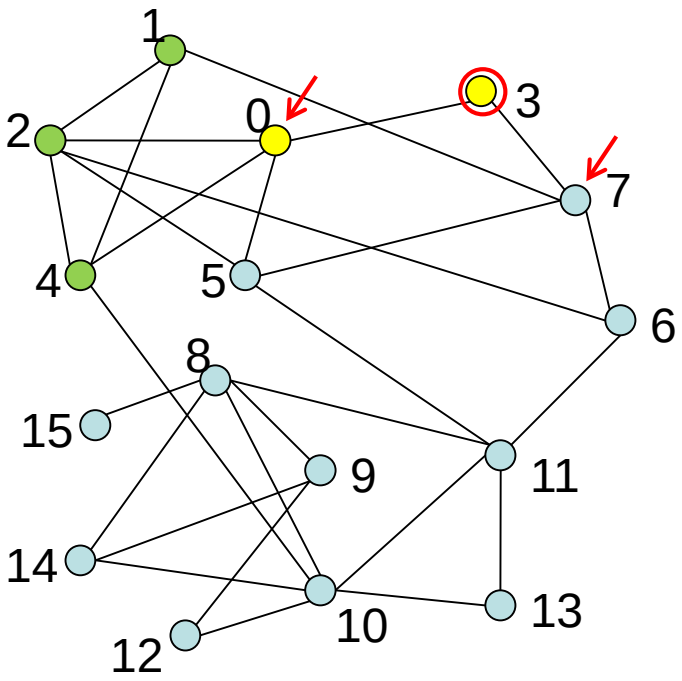
Pass 1 – Iteration 1
insert 0 in c[3]
insert 1 in c[4]

Louvain method – an example



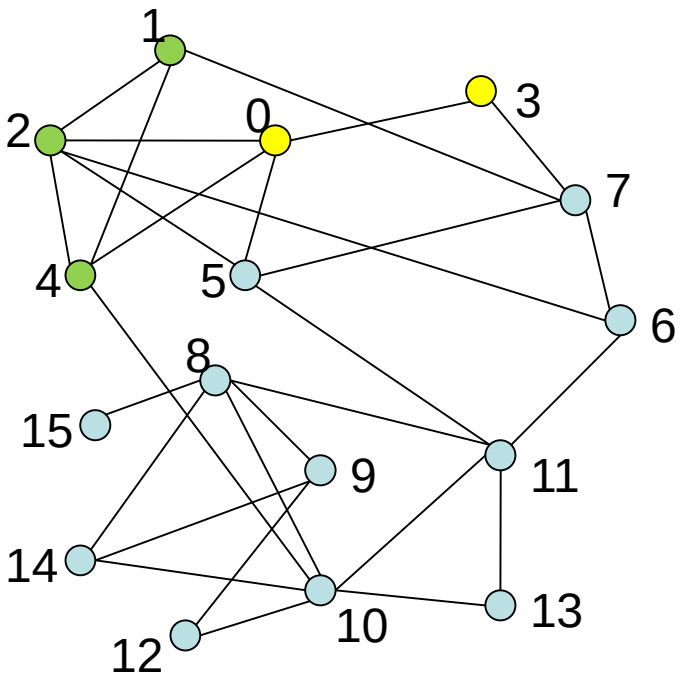
Pass 1 – Iteration 1
insert 0 in c[3]
insert 1 in c[4]
insert 2 in c[1,4]

Louvain method – an example



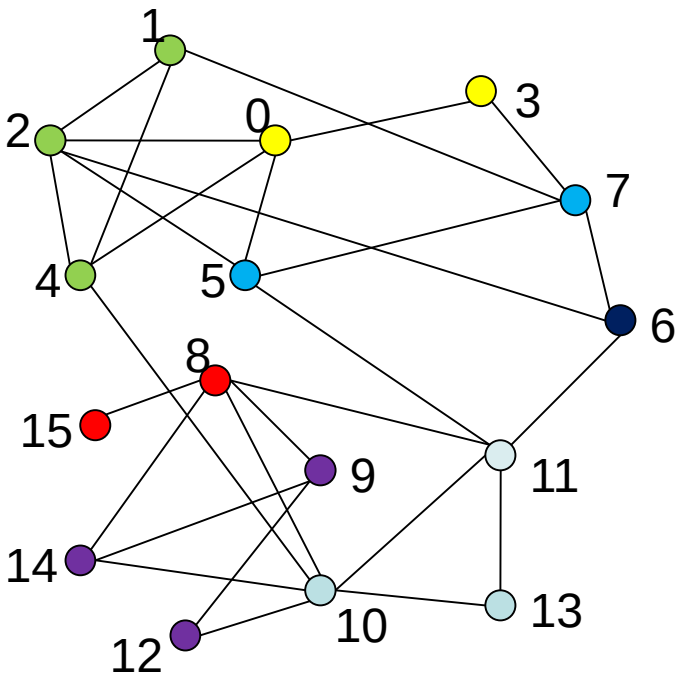
Pass 1 – Iteration 1
insert 0 in c[3]
insert 1 in c[4]
insert 2 in c[1,4]

Louvain method – an example



Pass 1 – Iteration 1
insert 0 in c[3]
insert 1 in c[4]
insert 2 in c[1,4]
insert 3 in c[0]

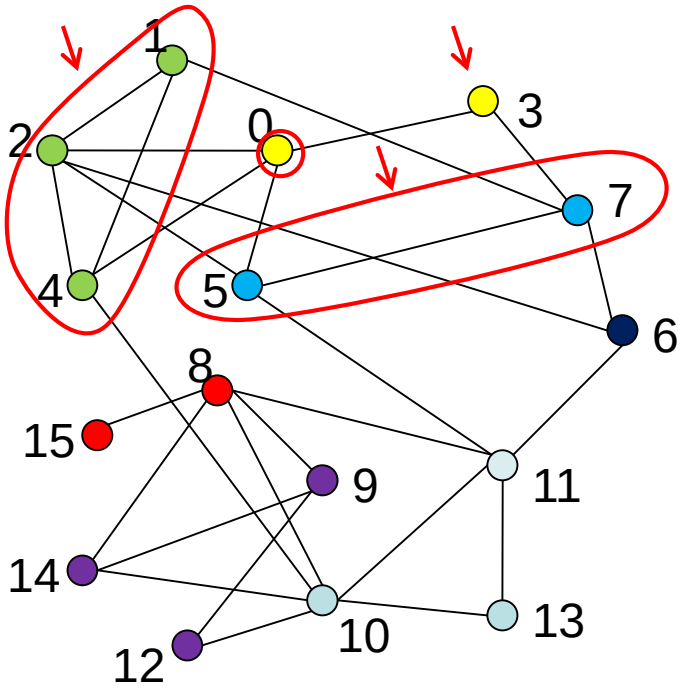
Louvain method – an example



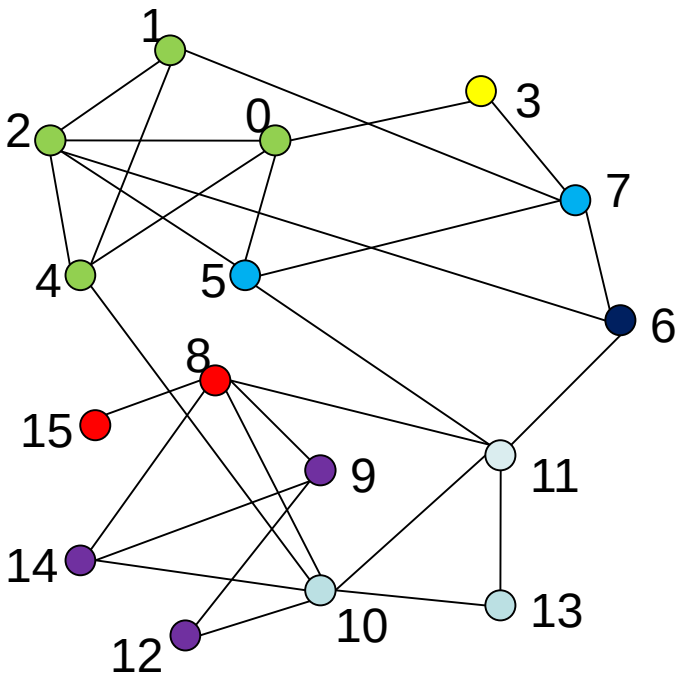
Pass 1 – Iteration 1
insert 0 in c[3]
insert 1 in c[4]
insert 2 in c[1,4]
insert 3 in c[0]
insert 4 in c[1]
insert 5 in c[7]
insert 6 in c[11]
insert 7 in c[5]
insert 8 in c[15]
insert 9 in c[12]
insert 10 in c[13]
insert 11 in c[10,13]
insert 12 in c[9]
insert 13 in c[10,11]
insert 14 in c[9,12]
insert 15 in c[8]

Louvain method – an example

Pass 1 – Iteration 2

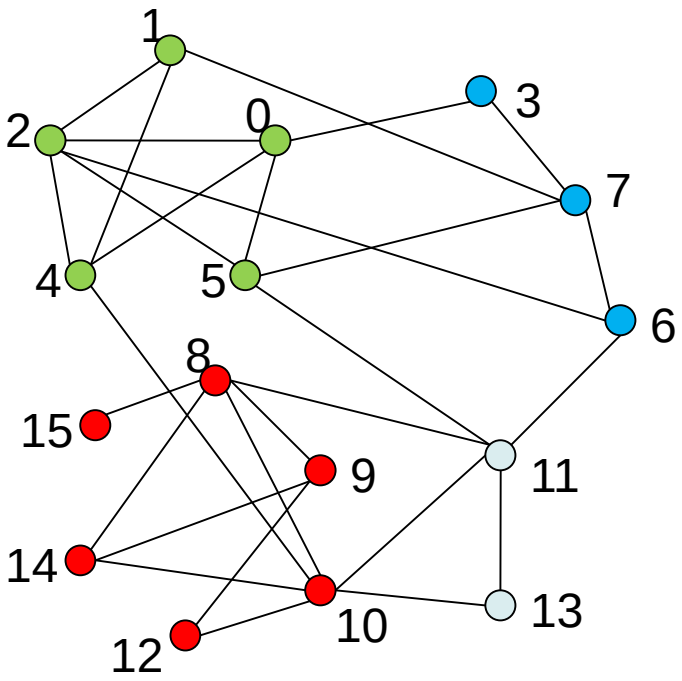


Louvain method – an example



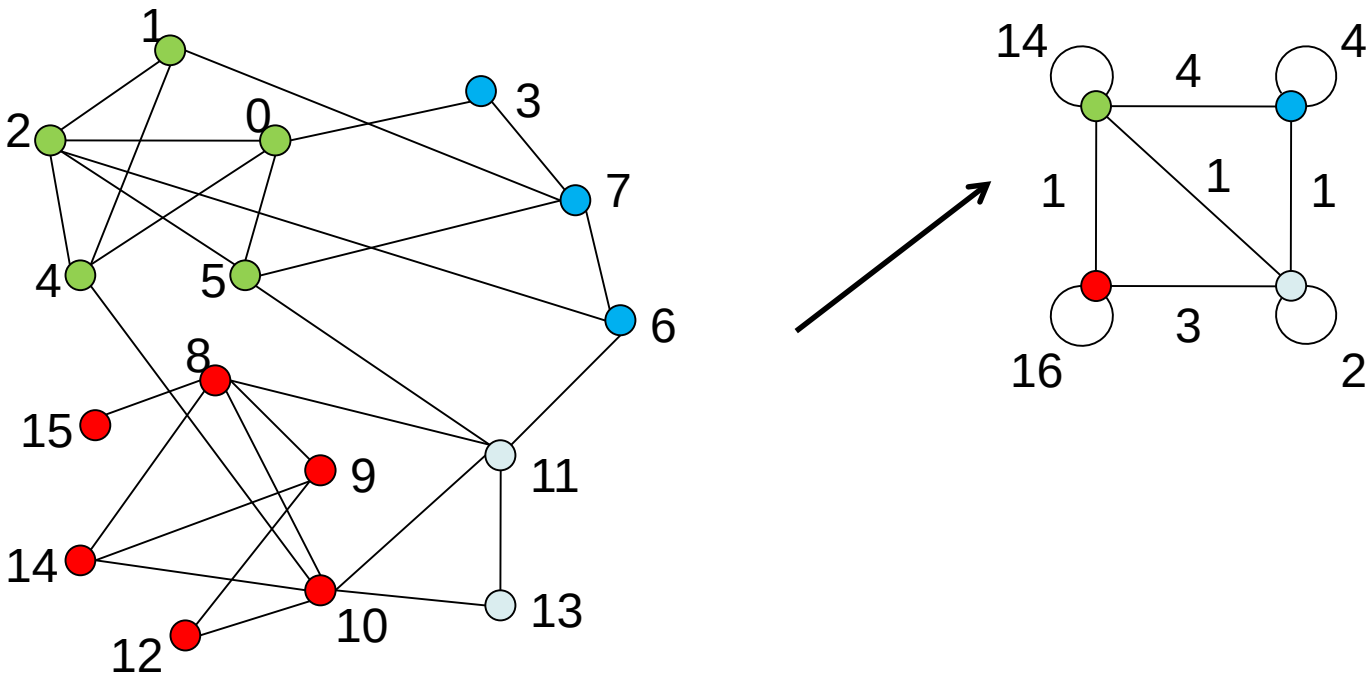
Pass 1 – Iteration 2
insert 0 in c[4]
…

Louvain method – an example

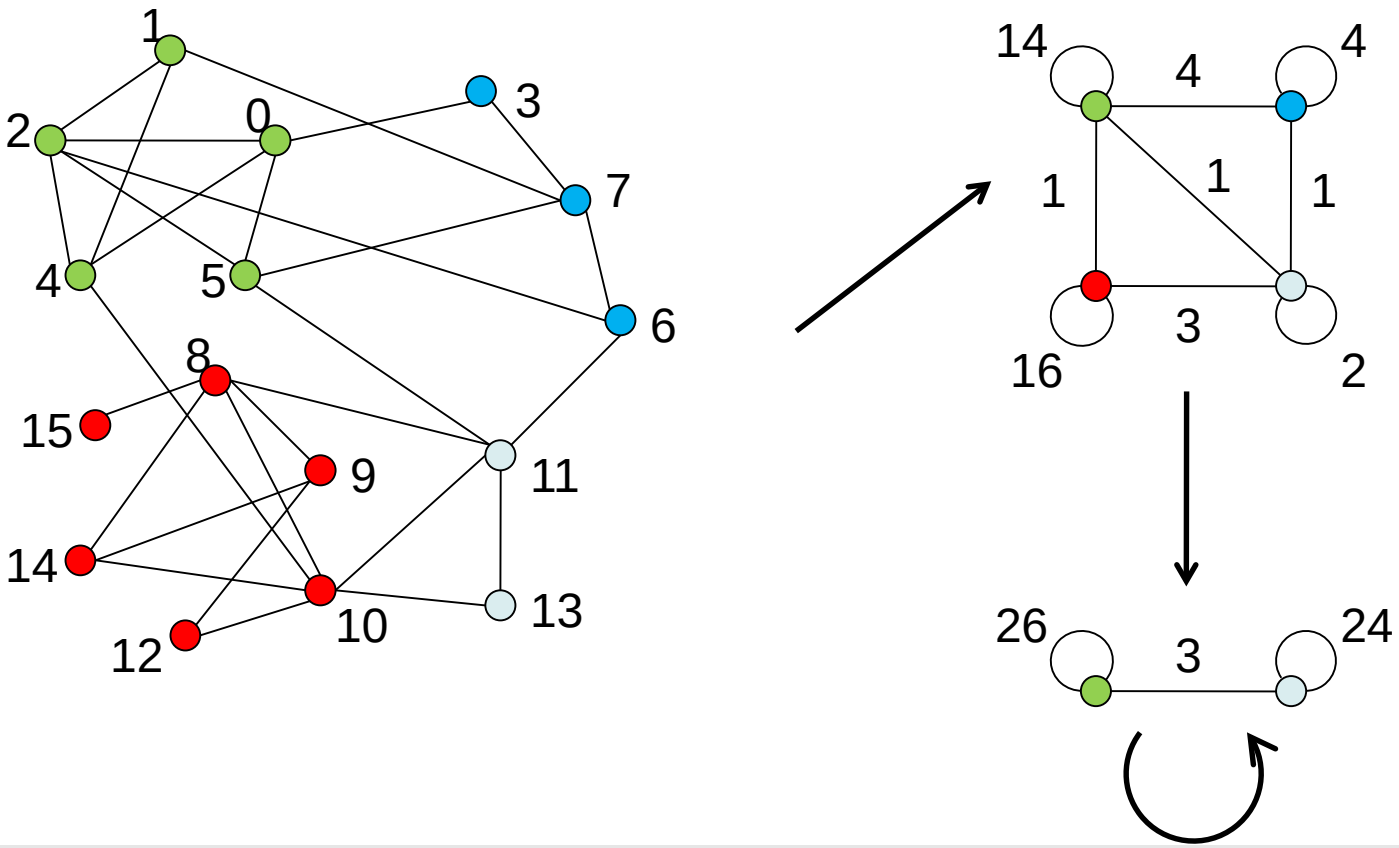


After 4 iterations, local maxima is reached

Louvain method – an example



Louvain method – an example



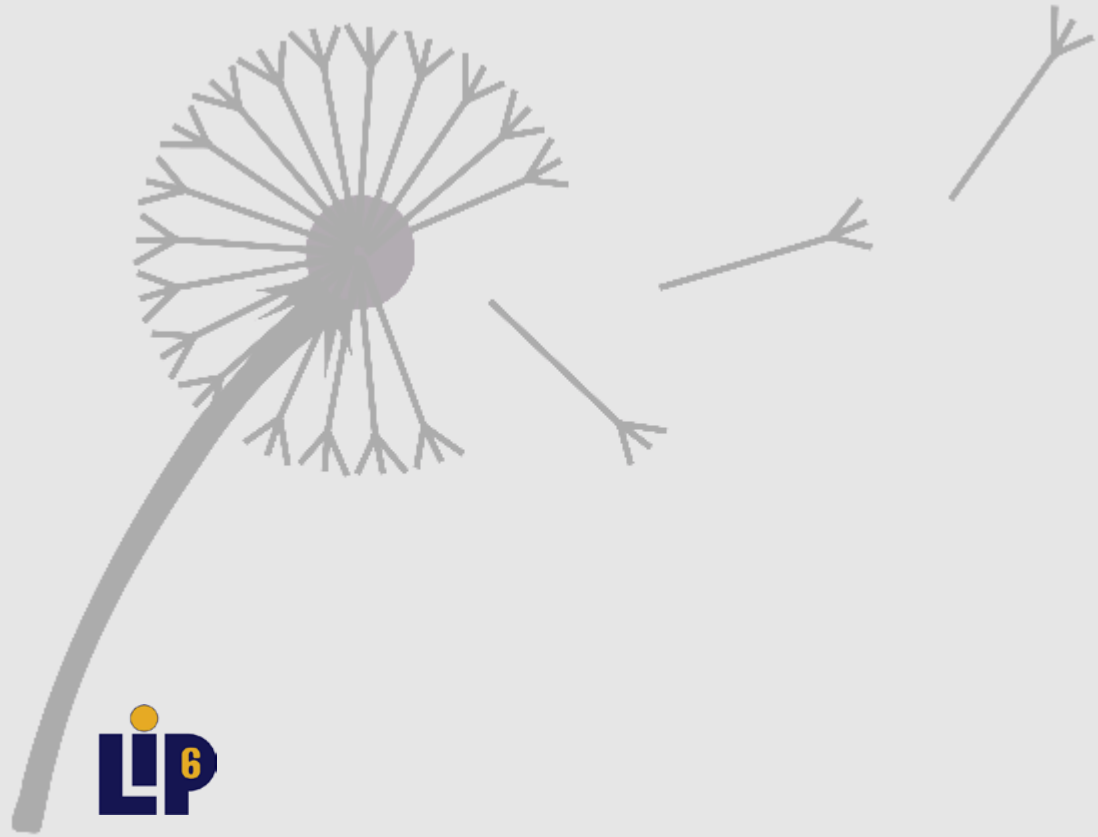
Experimental results (time)

	Karate	Arxiv	Internet	Web nd.edu	Belgian Phone Calls	Web UK-2005	Web Webbase01
	n=34/m=77	9k/24k	70k/351k	325k/1M	2.5M/6.3M	39M / 783M	118M/1B
Newman Girvan Clauset Moore	0s	3.6s	799s	5034s			
Pons Latapy	0s	3.3s	575s	6666s			
Wakita Tsurumi (expected)	0s	0s	8s	52s	1279s	(3days)	
Louvain	**0s**	**0s**	**<1s**	**<1s**	**47s**	**252s**	**469s**
	3 passes	5 passes	5 passes	5 passes	5 passes	4 passes	5 passes

Experimental results (Q)

	Karate	Arxiv	Internet	Web nd.edu	Belgian Phone Calls	Web UK-2005	Web Webbase01
	n=34/m=77	9k/24k	70k/351k	325k/1M	2.5M/6.3M	39M / 783M	118M/1B
Newman Girvan Clauset Moore	0s 0.38	3.6s 0.772	799s 0.692	5034s 0.927			
Pons Latapy	0s 0.42	3.3s 0.757	575s 0.729	6666s 0.895			
Wakita Tsurumi (expected)	0s	0s	8s	52s	1279s	(3days)	
Louvain	**0s** **0.42**	**0s** **0.813**	**<1s** **0.781**	**<1s** **0.935**	**47s** **0.769**	**252s** **0.979**	**469s** **0.984**
	3 passes	5 passes	5 passes	5 passes	5 passes	4 passes	5 passes

COMMUNITY DETECTION THROUGH CONSENSUS



LIP6

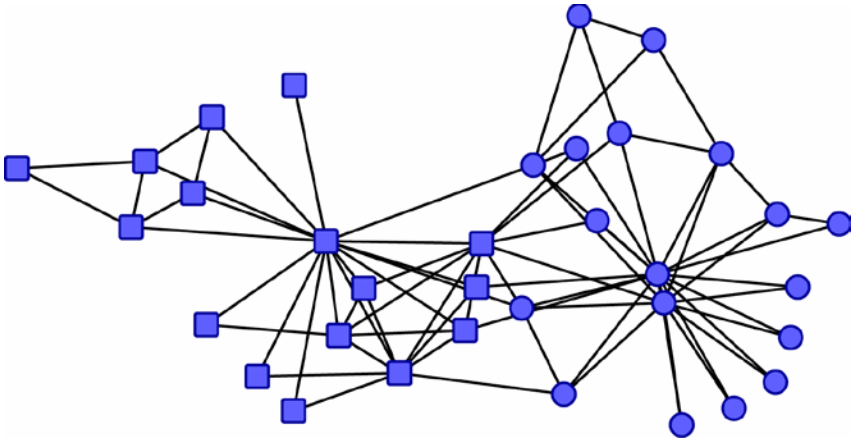
UPMC SORBONNE UNIVERSITÉS

cnrs dépasser les frontières

Consensual communities

- For a given graph:
  - Many high quality partitions
  - Are there similarities between these partitions?

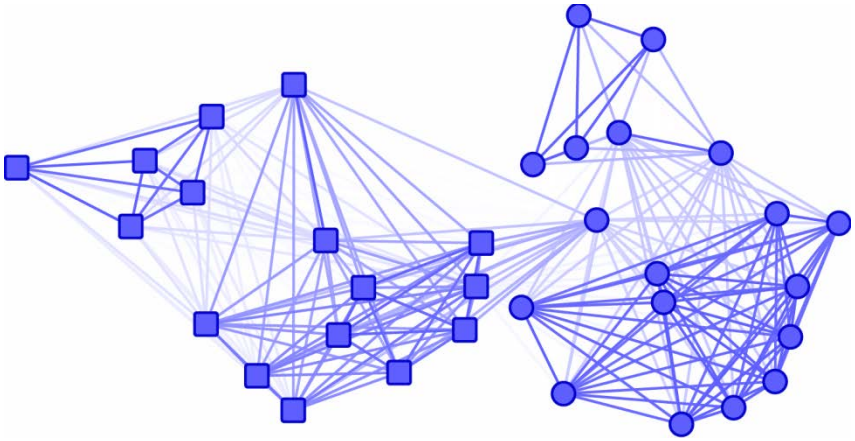
- Preliminary results:
  - Use Louvain method:
    - Non deterministic -> ≠ partitions.



Principe

- For a given graph:
  - Many high quality partitions
  - Are there similarities between these partitions?

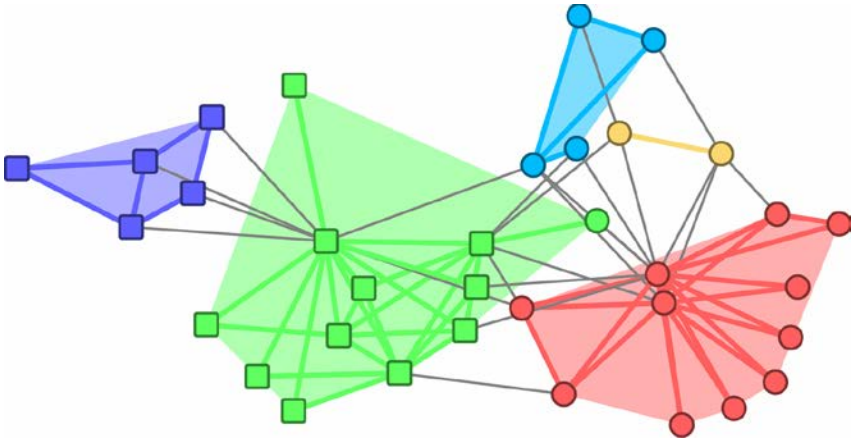
- Preliminary results:
  - Use Louvain method:
    - Non deterministic -> ≠ partitions.
  - Similarity graph :
    - Proximity = similarities between partitions.



Consensual communities

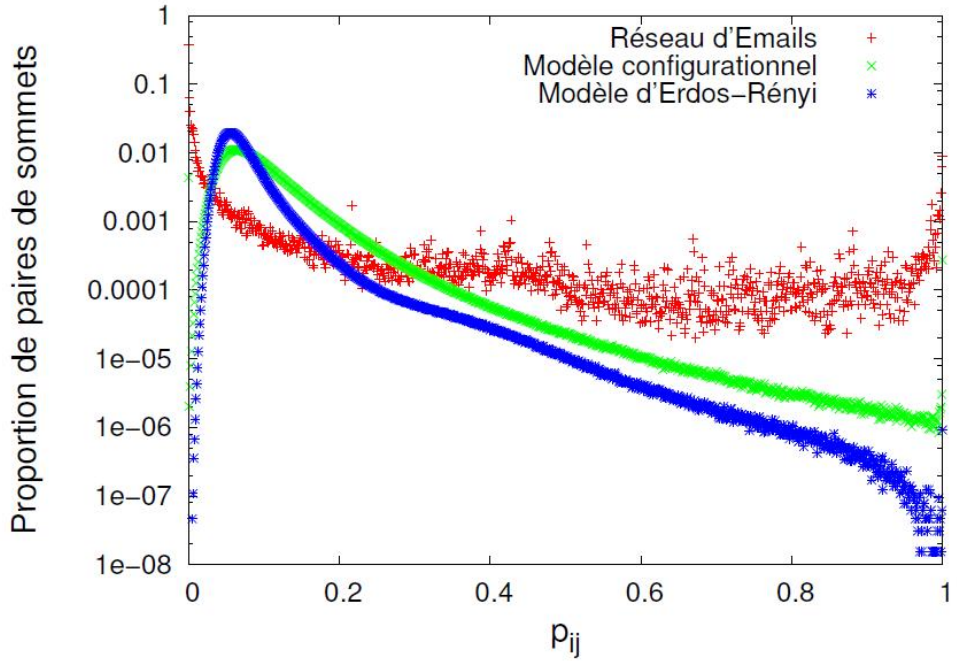
- For a given graph:
  - Many high quality partitions
  - Are there similarities between these partitions?

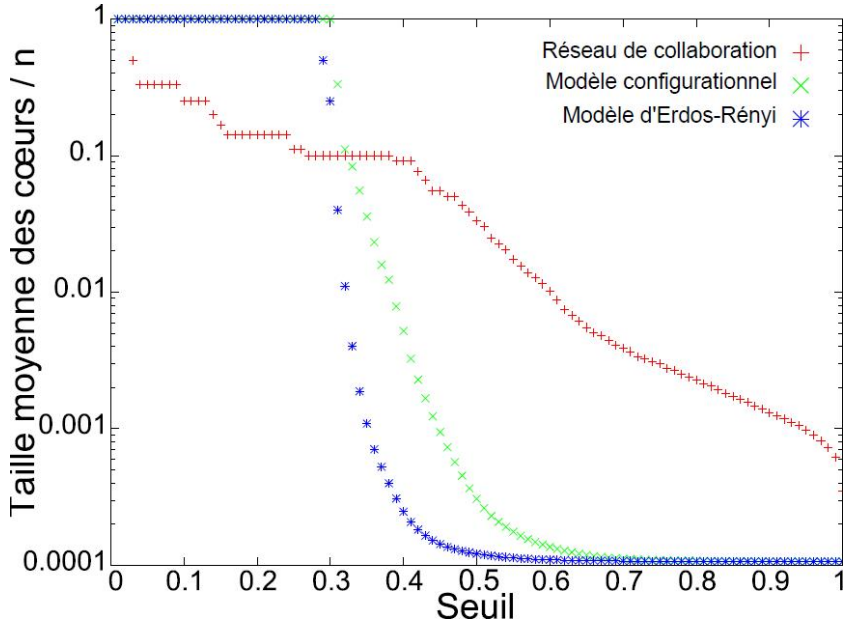
- Preliminary results:
  - Use Louvain method:
    - Non deterministic -> ≠ partitions.
  - Similarity graph :
    - Proximity = similarities between partitions.
  - Threshold on the proximity.



Existence of consensual communities

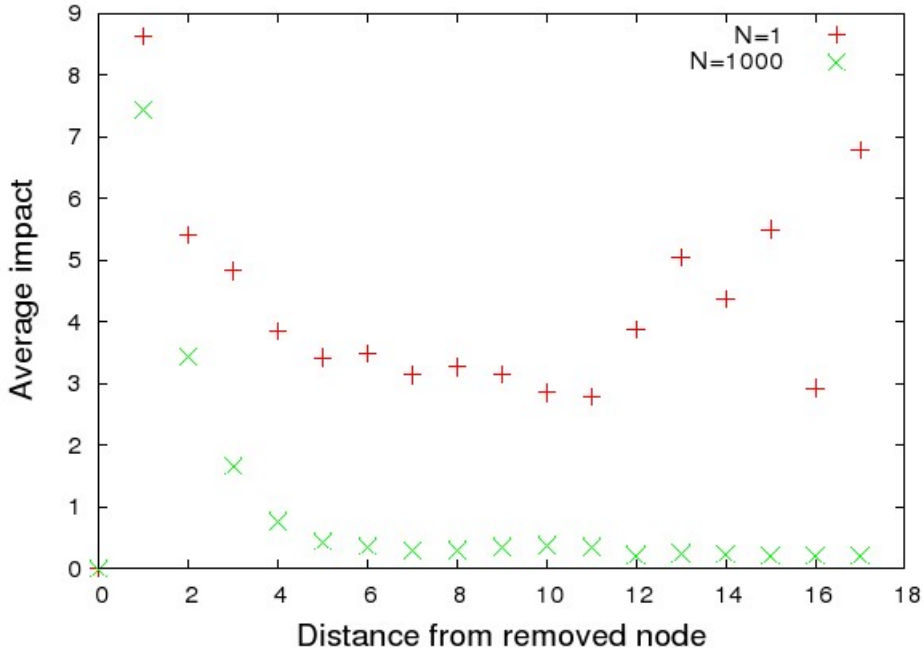
- In real data, many similarities
- Not in random graphs





Impact locality

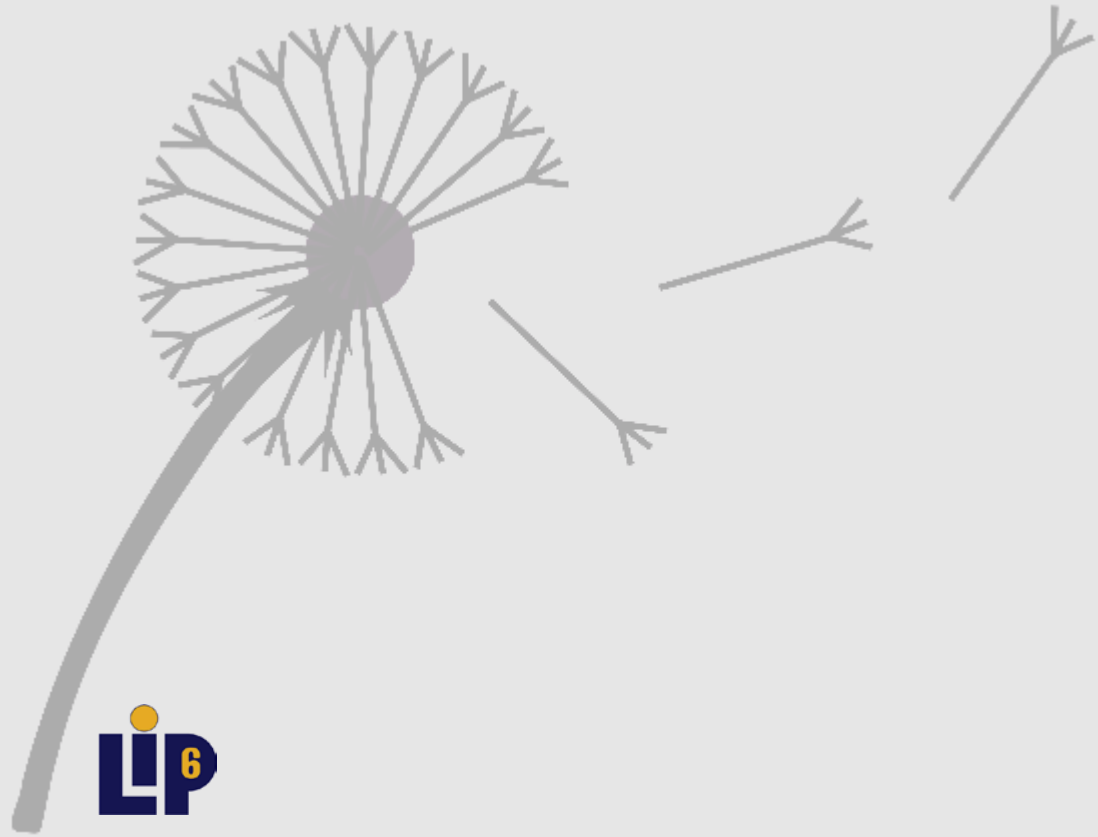
- Removal of one single node
- Impact on the community structure vs distance:
  - Communities: poorly related to distance
  - Consensual communities: lower impact and more related to distance



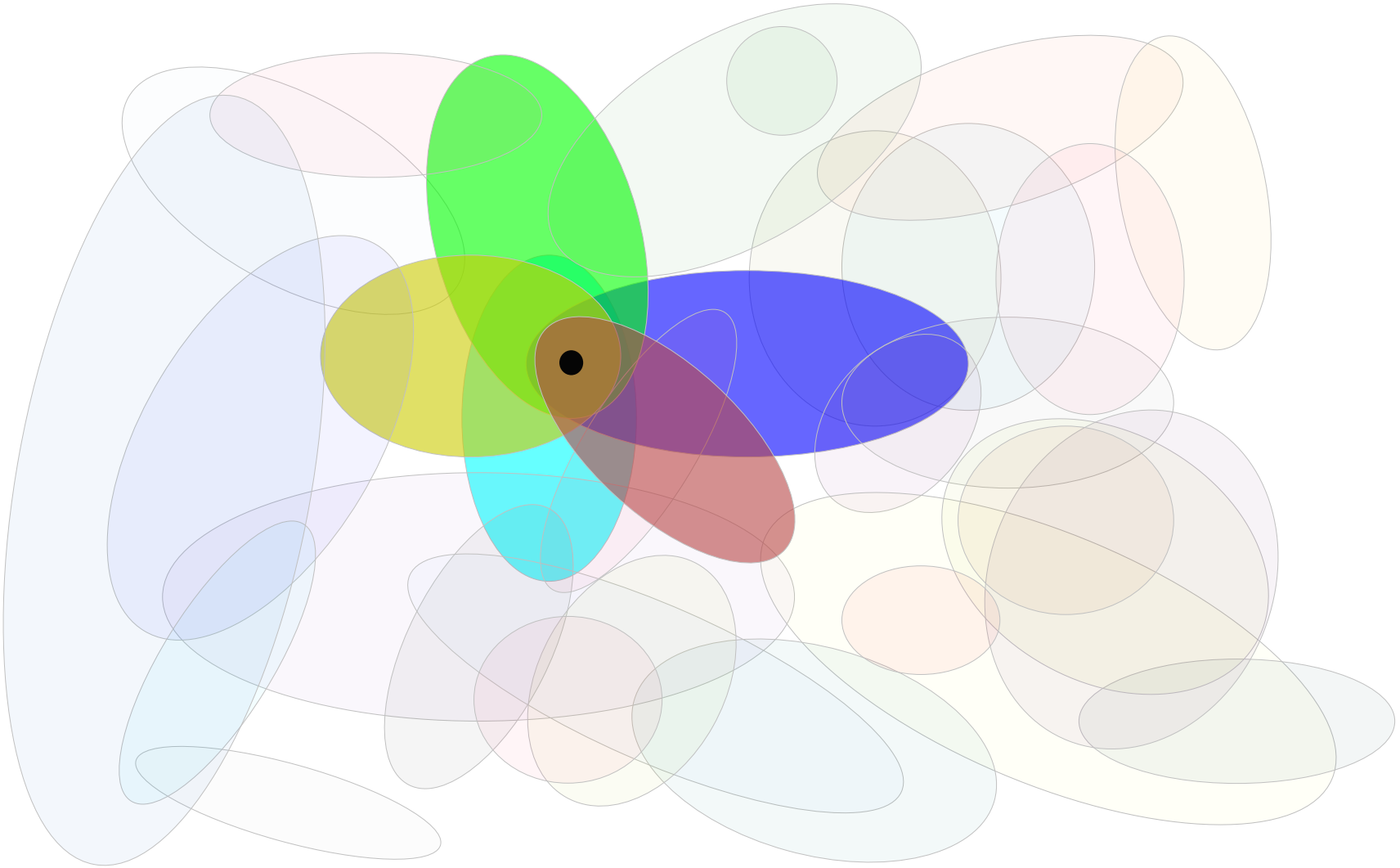
EGOCENTERED COMMUNITIES PARAMETER FREE MEASURE

IJWBC 2013
CompleNet 2013
SNAM 2014

Joint work with
Maximilien Danisch
Bénédicte Le Grand



Egocentered communities



Carryover opinion

Principle: information may be trapped in communities

- Proximity measure based on opinion dynamics
  - Node of interest have a constant opinion of 1
  - Each node takes the average opinion of its neighbors
- Similar to random walk approaches
  - But parameter-free and fast convergence

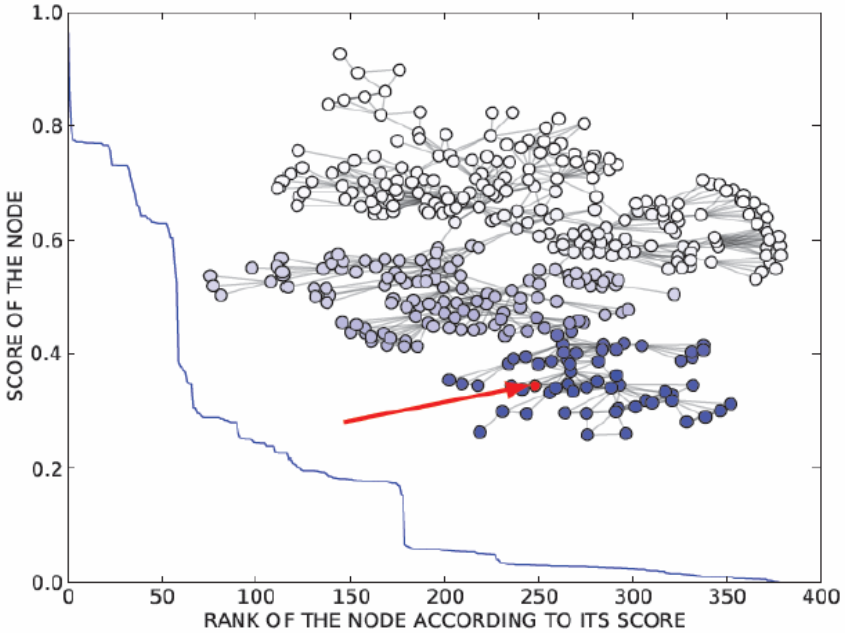
$$X_t = MX_{t-1} \qquad \text{AVERAGING}$$

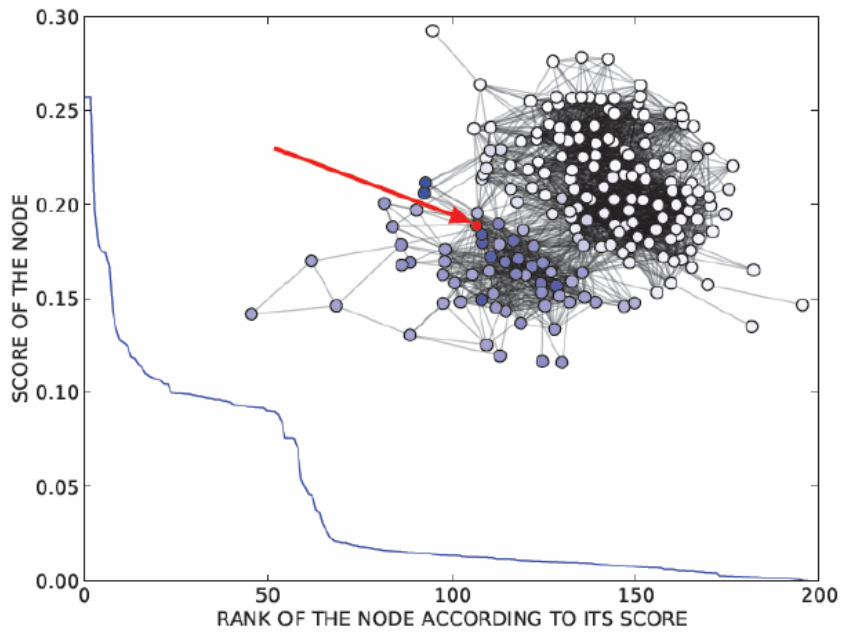
$$X_t = \frac{X_t - \min(X_t)}{1 - \min(X_t)} \qquad \text{RESCALING}$$

$$X_t^i = 1 \qquad \text{RESETTING}$$

Carryover opinion - examples

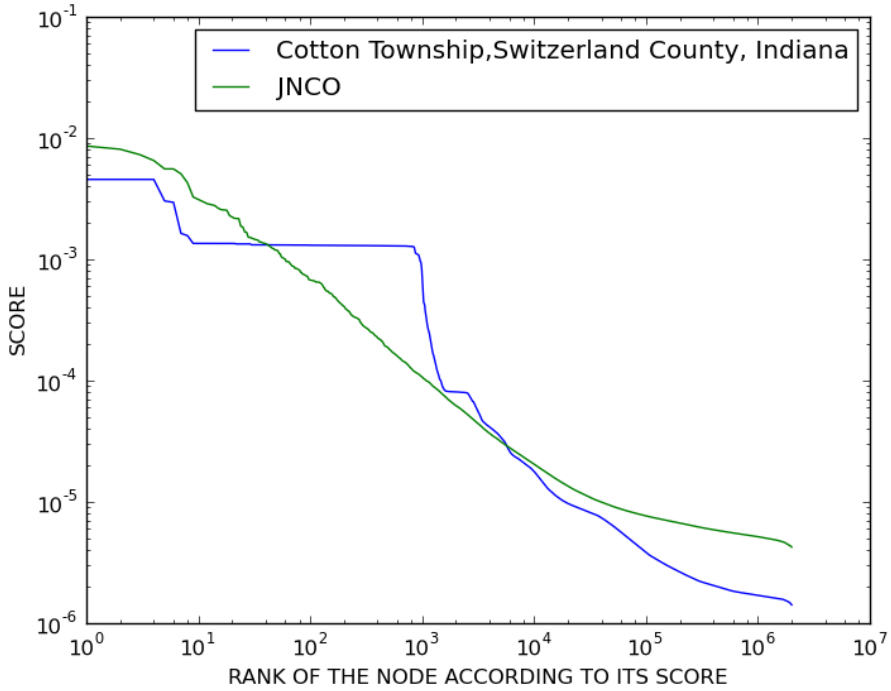
- Proximity measure based on opinion dynamics
  - Plateaux structure appears which corresponds to communities



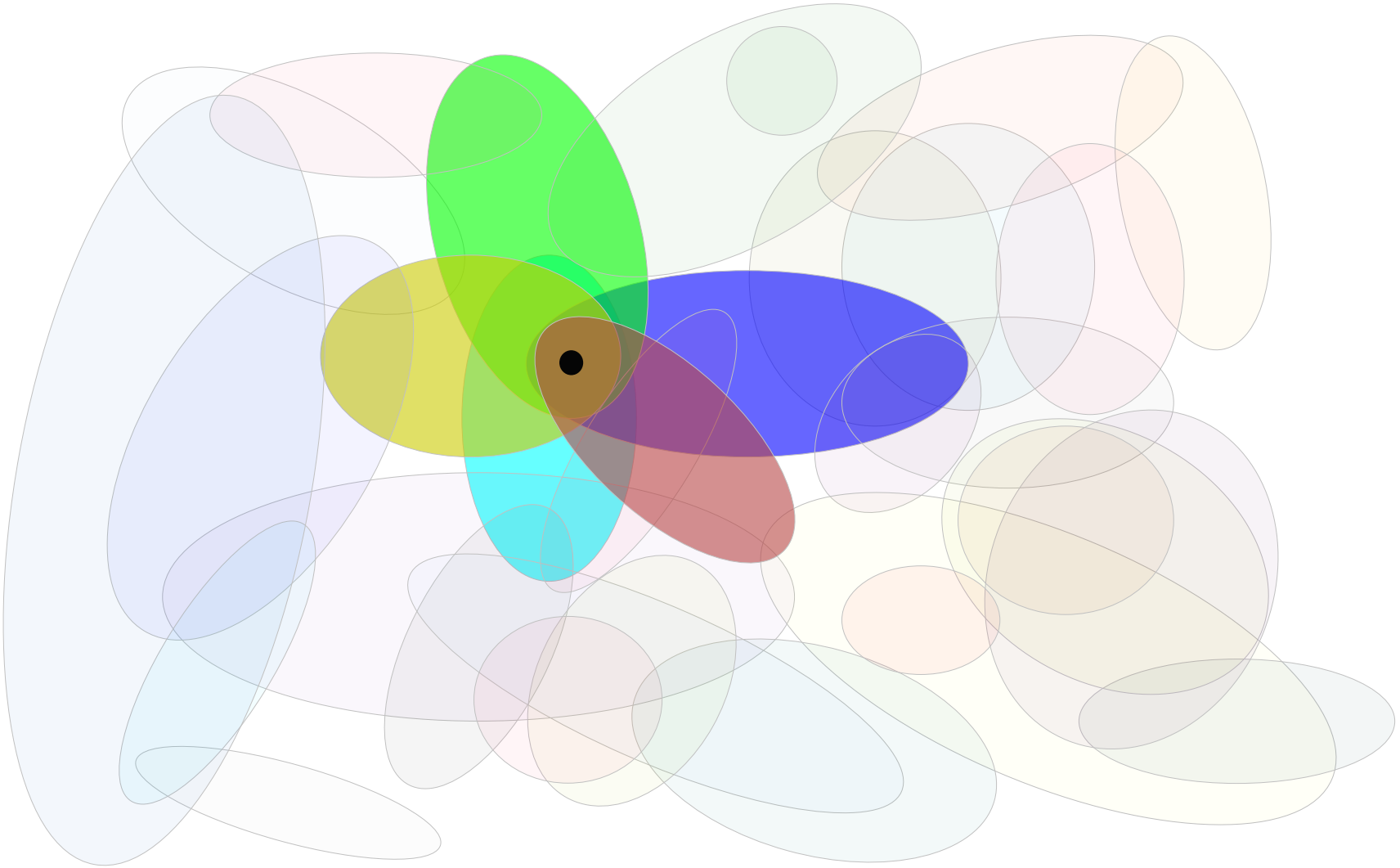


Carryover - limitations

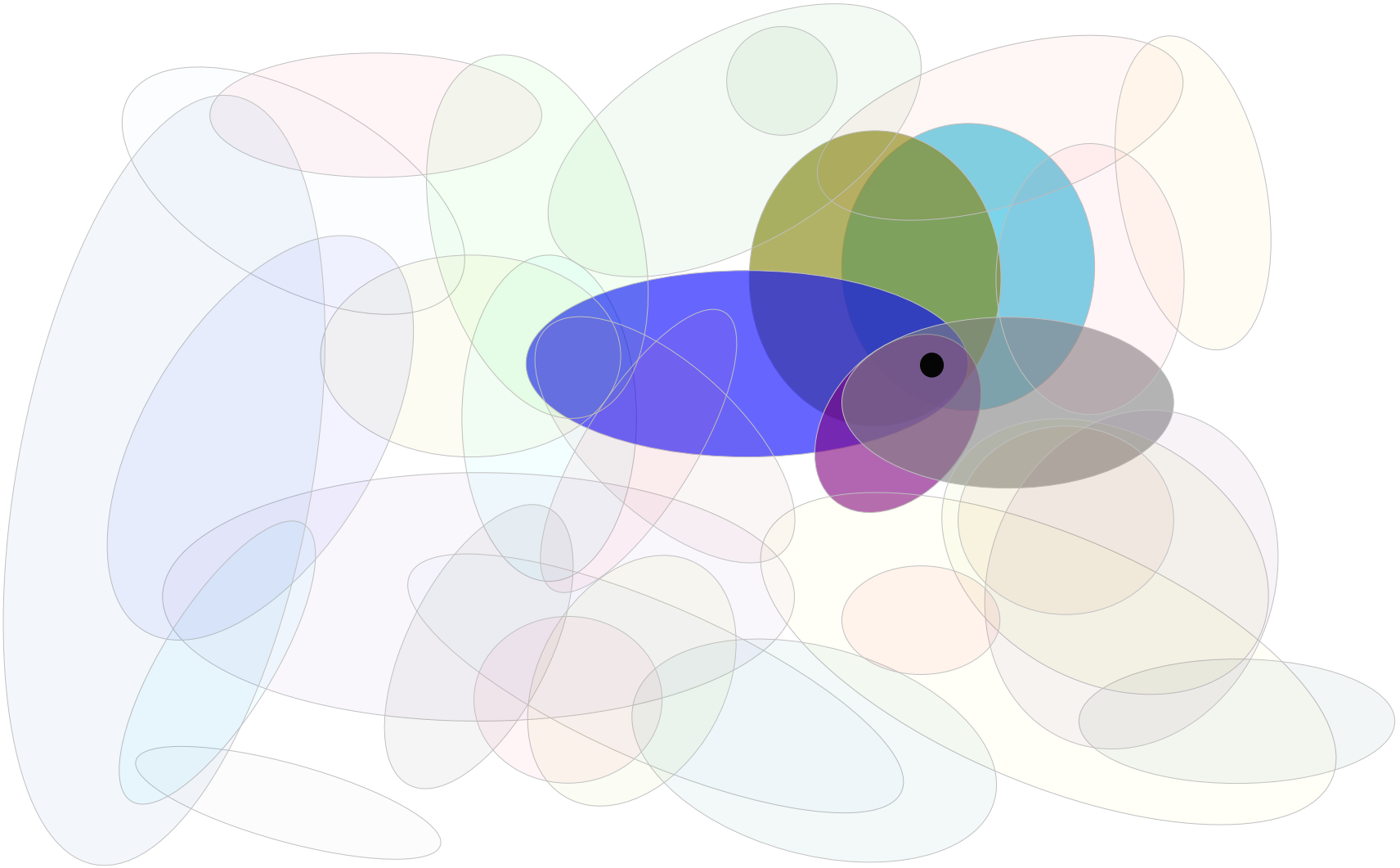
- More than one community containing the node:
  - Communities at different scales
  - Overlapping communities



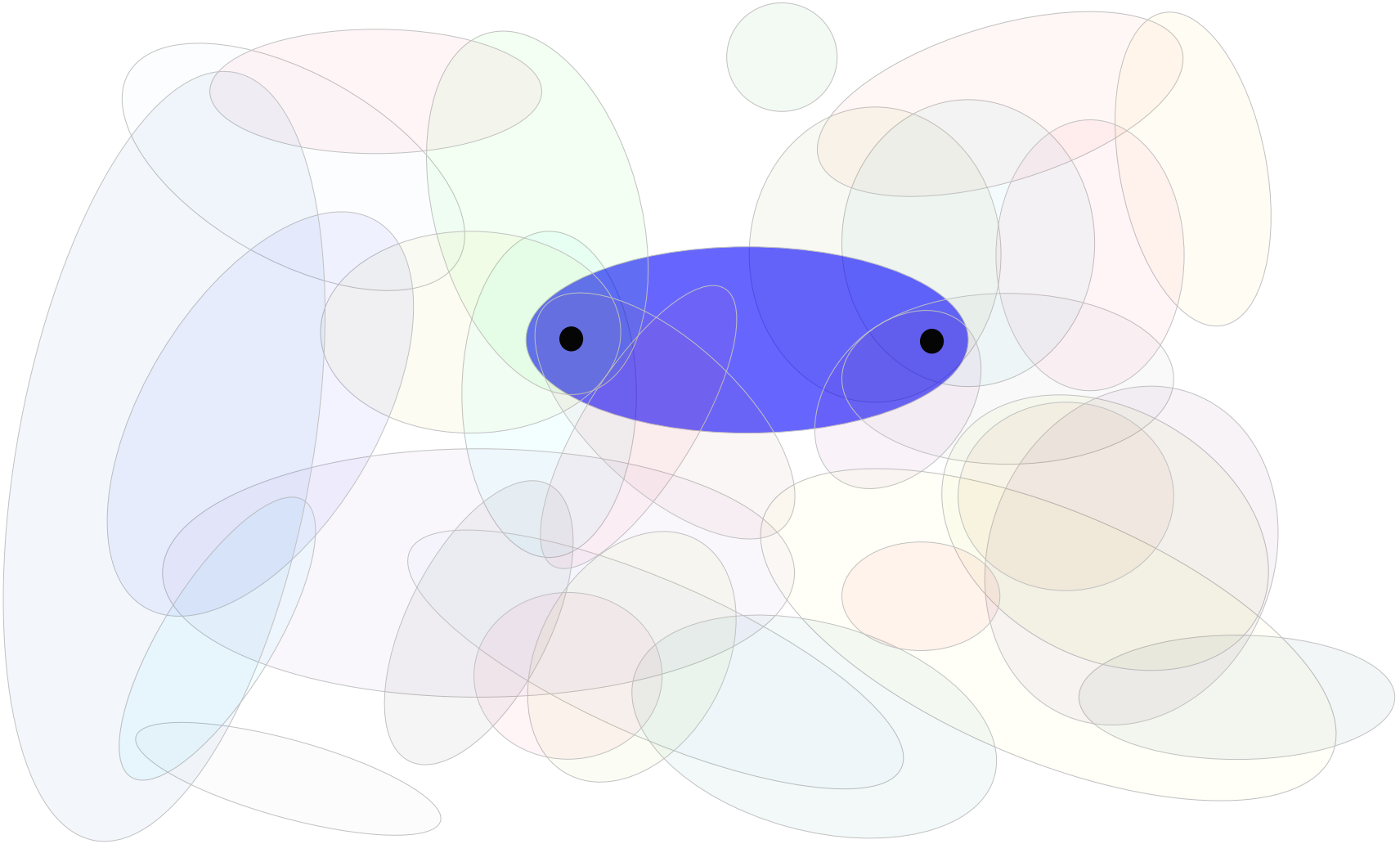
Egocentered communities



Egocentered communities



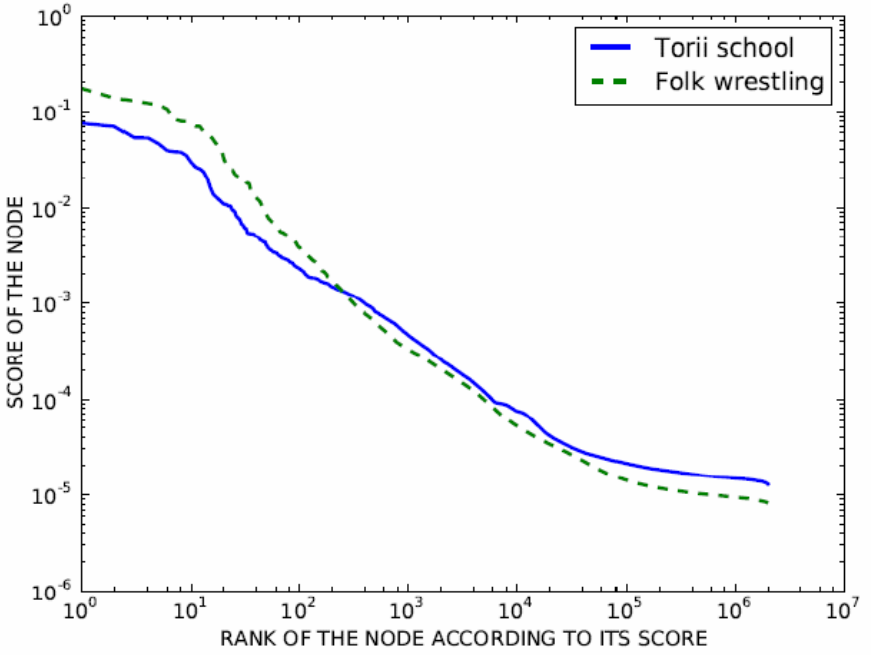
Egocentered communities

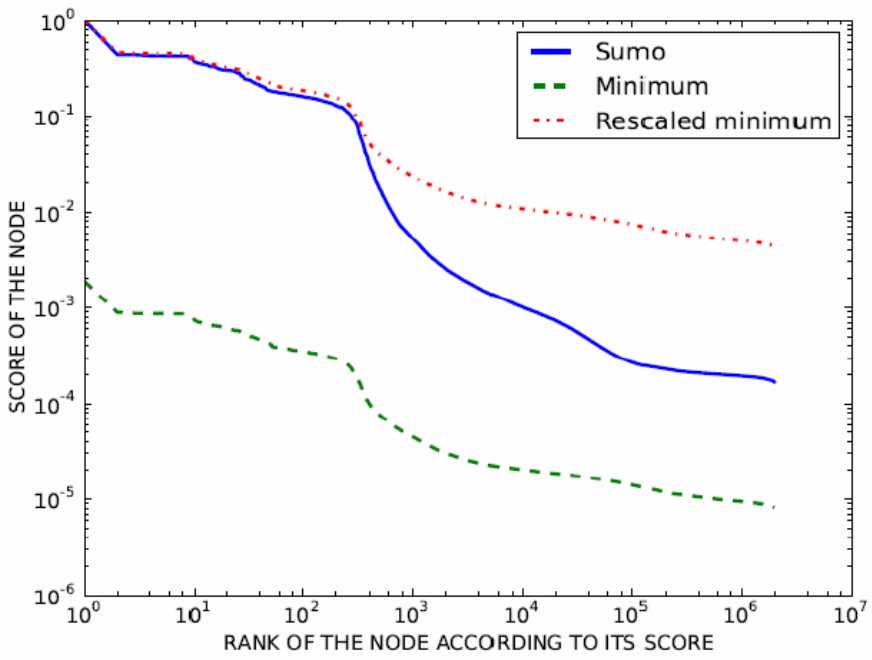


Bi-egocentered communities

Torii school + Folk wrestling = Sumo

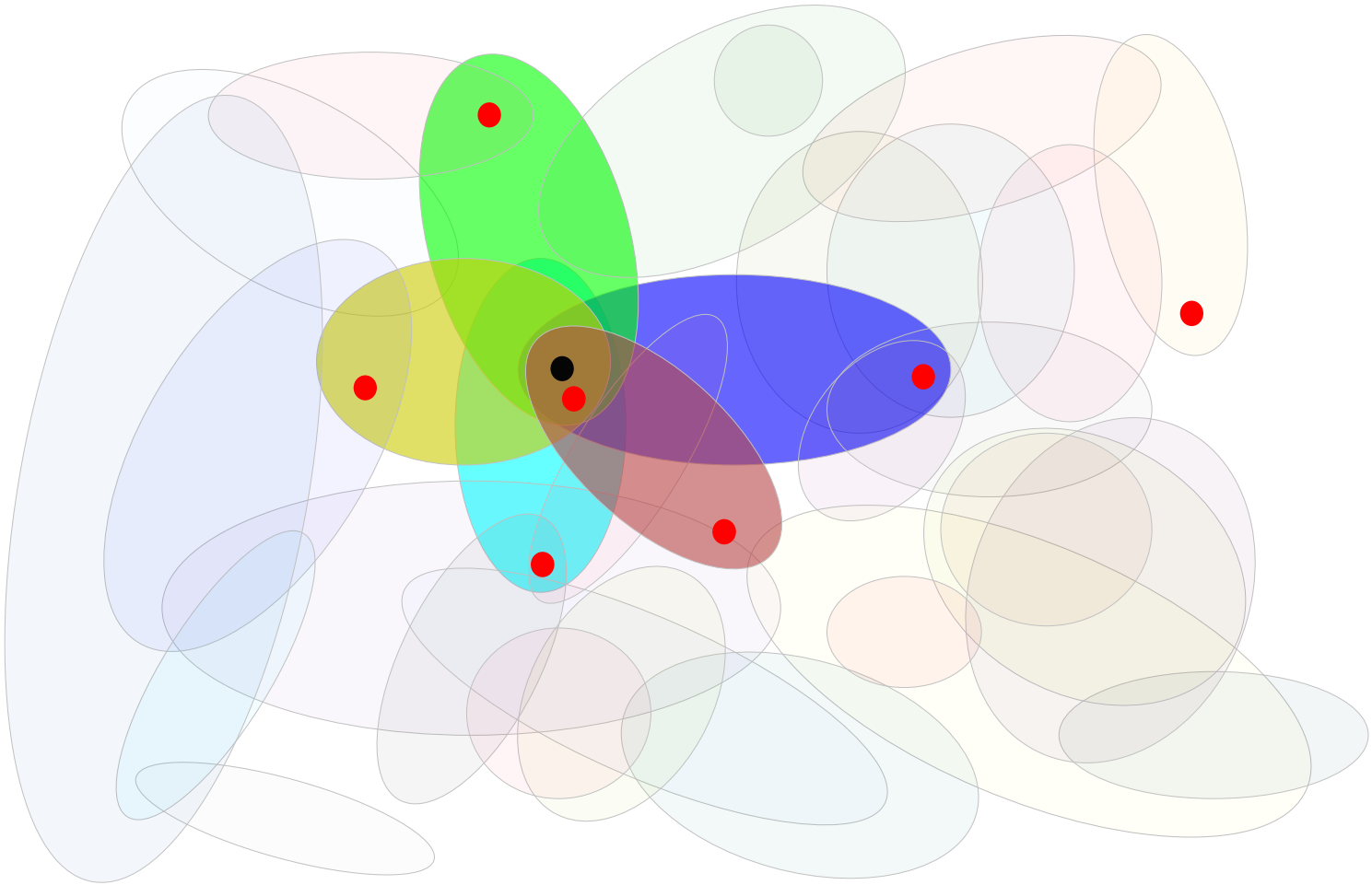
(350 first nodes of sumo contains 337 of the minimum)





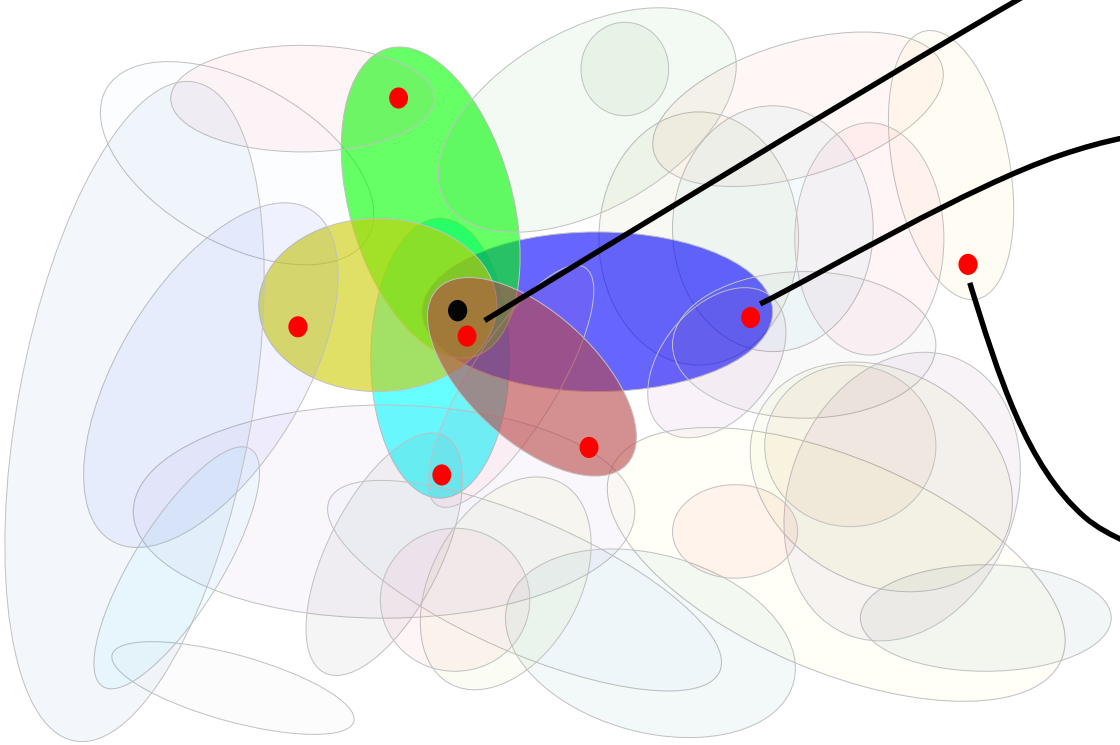
Methodology to find all communities

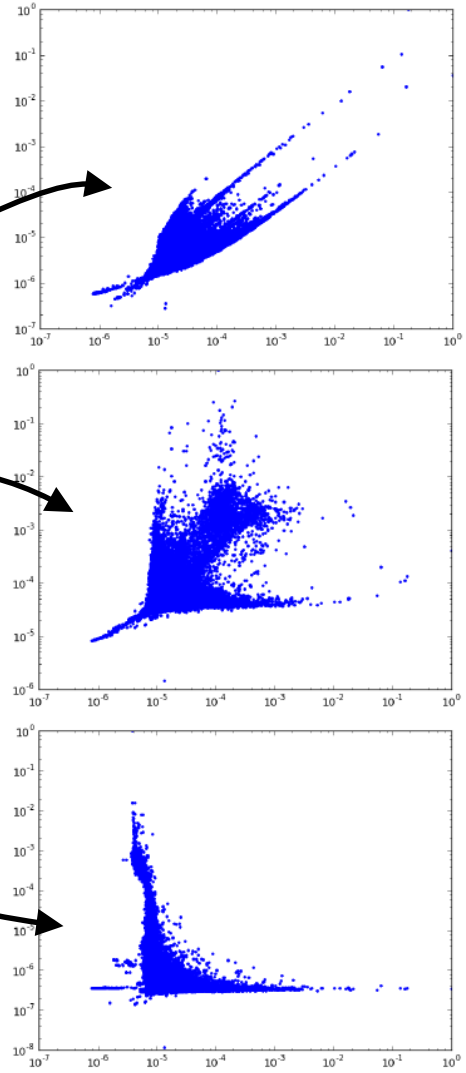
1. Select candidate nodes



Methodology to find all communities

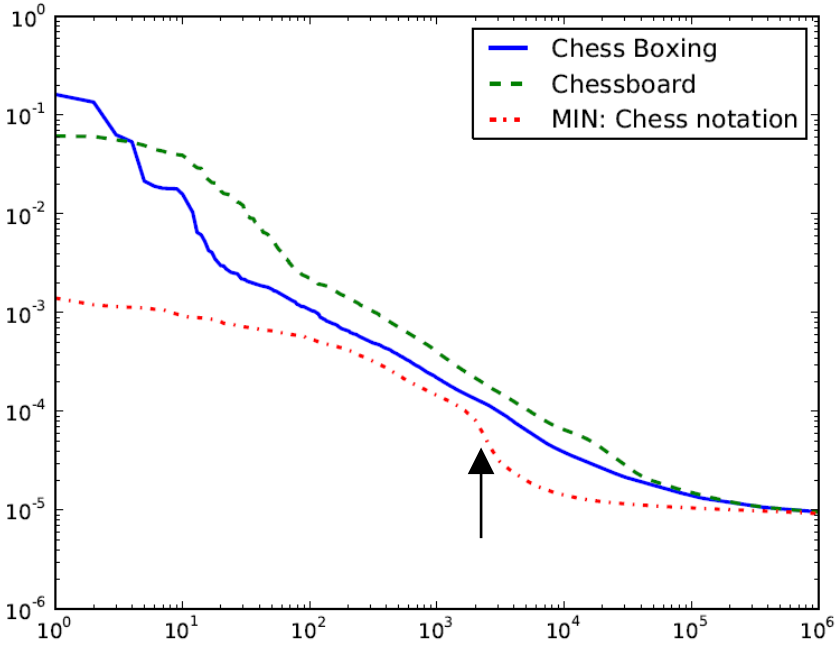
1. Select candidate nodes
   - Pick a random sample of intermediate nodes
   - Size of the sample / computation time

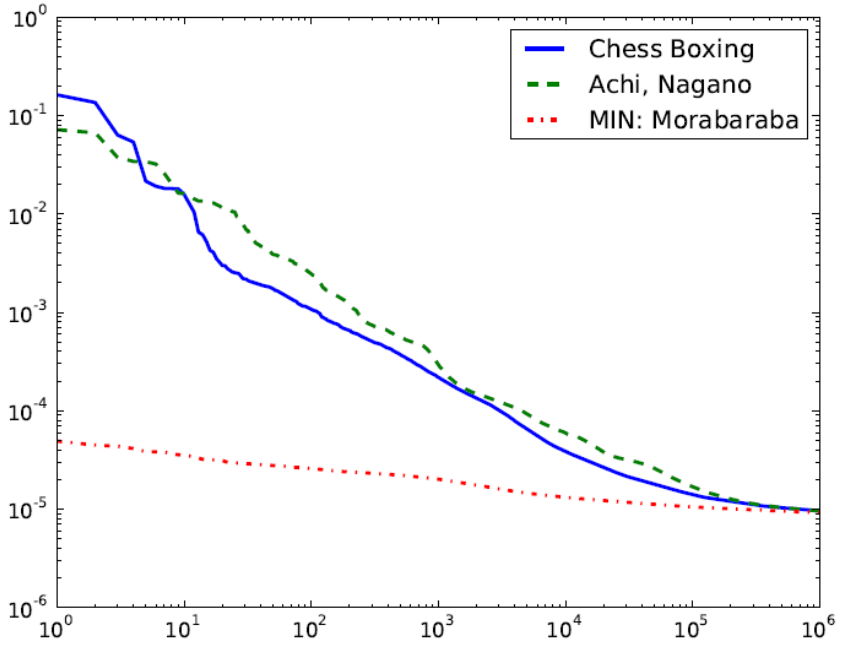




Methodology to find all communities

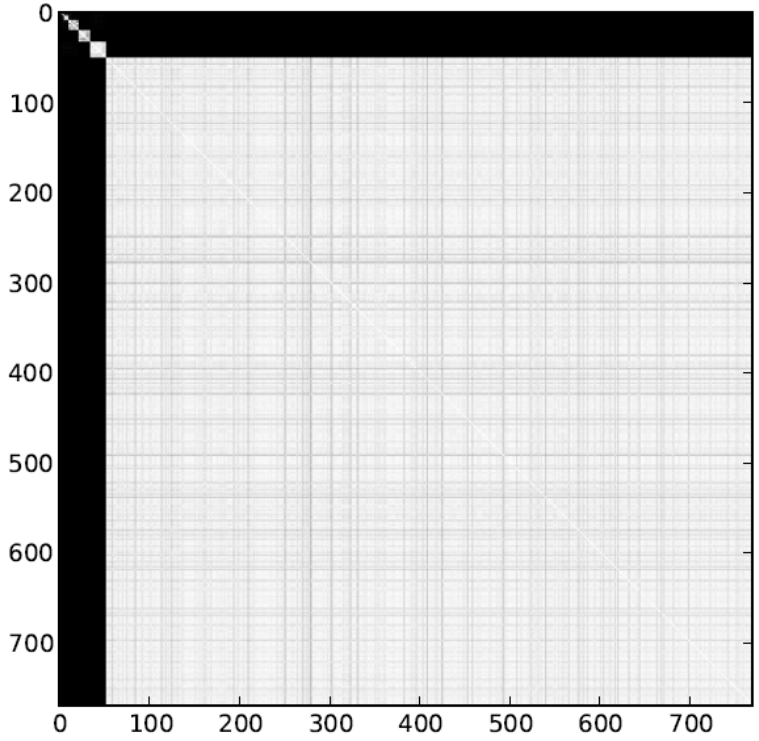
1. Select candidate nodes
2. Compute bi-egocentered communities
   - Minimum of the two scores
   - Keep nodes before the sharp decrease (if any)

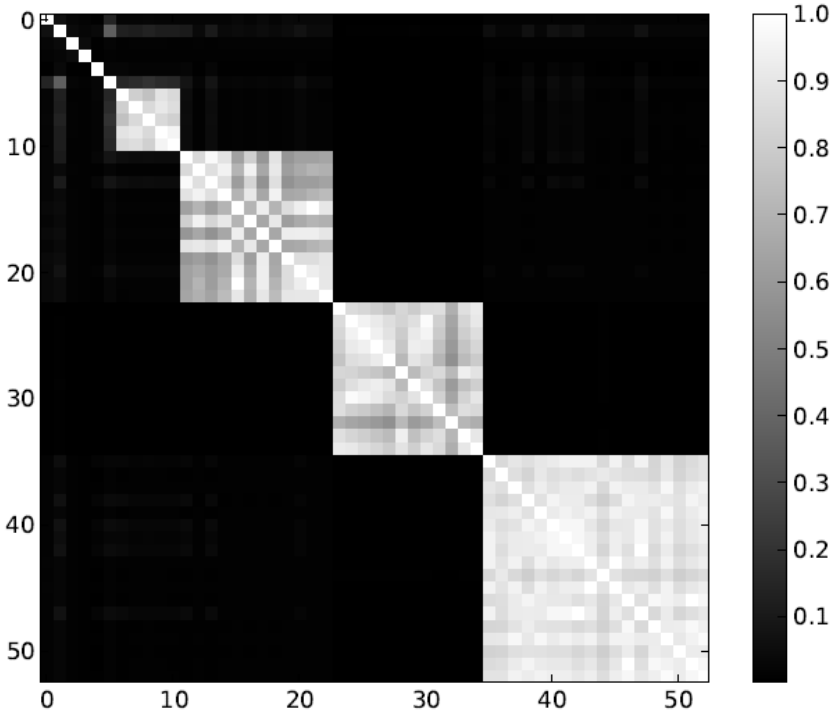




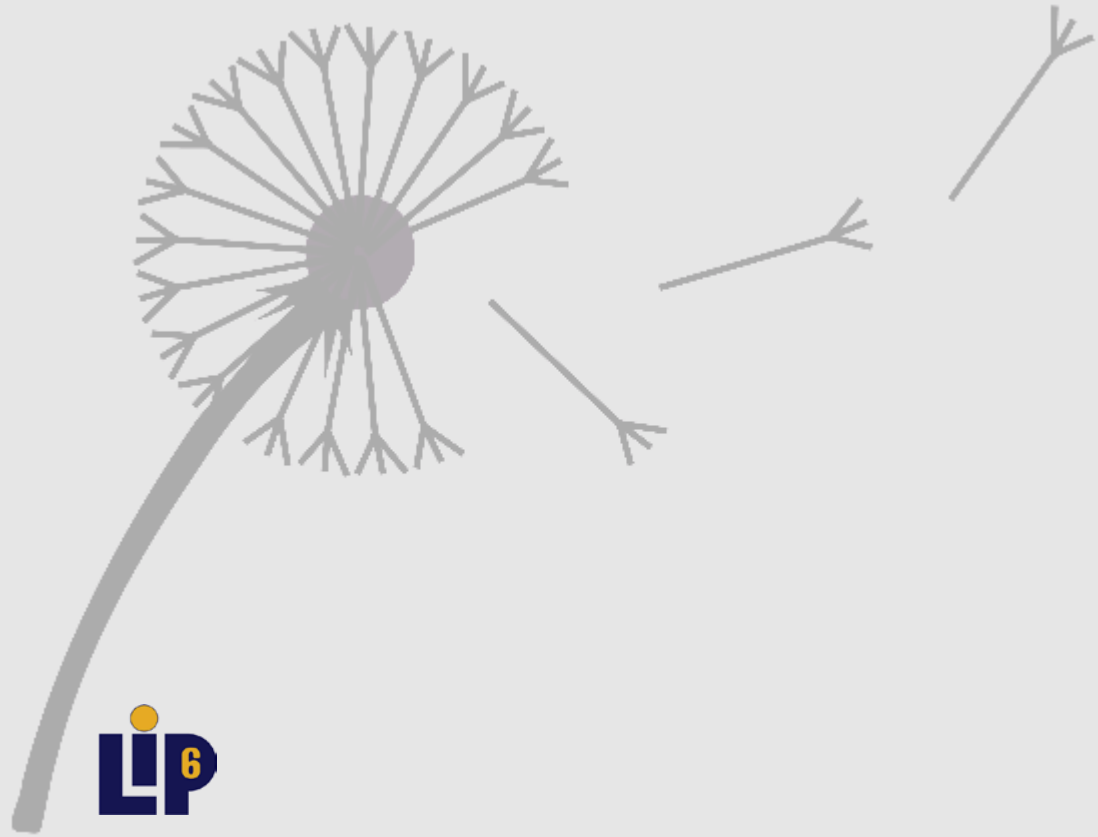
Methodology to find all communities

1. Select candidate nodes
2. Compute bi-egocentered communities
3. Clean output: merge similar and remove unique communities





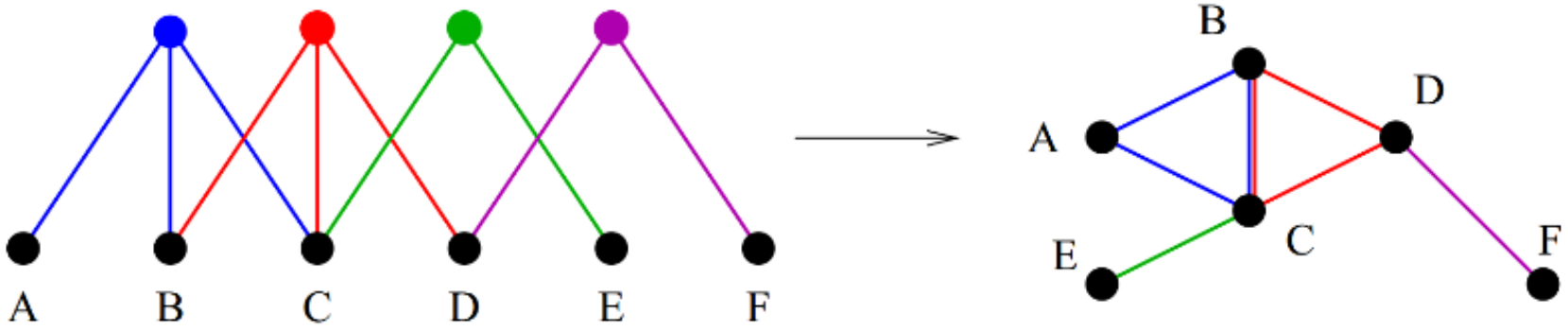
More problems



Bipartite decomposition

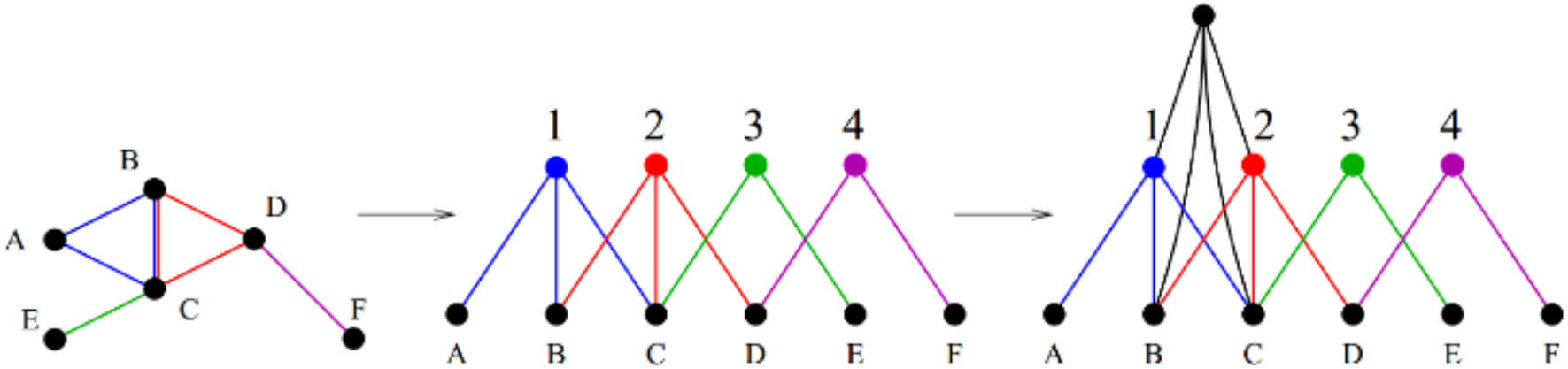
- Many networks are naturally bipartite
  - One-mode projection : top nodes -> cliques
  - Model to generate random bipartite graph

- Reverse operation :
  - Compute cliques -> bipartite graph
  - Covering of the graph by cliques?



Tripartite decomposition

- Real cliques are not randomly overlapping
  - How to estimate/compute this overlapping
  - Compute non trivial bipartite cliques
- Can be used for different, yet related, problems
  - Overlapping between consensual communities?



Counting triangles

- 41.652.230 nodes, 1.202.513.046 links, 34.824.916.864 triangles
  - Hadoop, 1636 nodes: 423 mn
  - GraphChi (Mac Mini, 8Go RAM): 60 mn
  - PowerGraph 64 nodes x 8 cores: 1.5 mn
    - http://www.bigdatarepublic.com/author.asp?section_id=2840&doc_id=269178
  - Sequential computation (ML algo) 10 Go: 59 mn
    - http://www-rp.lip6.fr/~latapy/Triangles/

- Basic principles:
  - For a node v, considering all pairs of neighbors is costly if v has high degree => start with low degree nodes
  - High and low degrees nodes can be considered separately