Community detection on real world networks: handling the dynamic, the overlap and the relevance of the results

Cazabet Rémy

Who am I ?

- University of Toulouse
- PhD Student (not for long)
- Computer Science

What is my thesis about ?

What is my thesis about ?

What is my thesis about ?

What is my thesis about ?

What is my thesis about ?

What is my thesis about ?

Complex, temporal networks

Multi-agent Approach

Dynamic community detection

Applications

Complex, temporal networks

Real, large networks

- Properties :
  - Very large (millions of nodes)
  - Small world, power law, clustering...
  - noisy, incomplete, ill-defined, ...
  - Evolving continuously

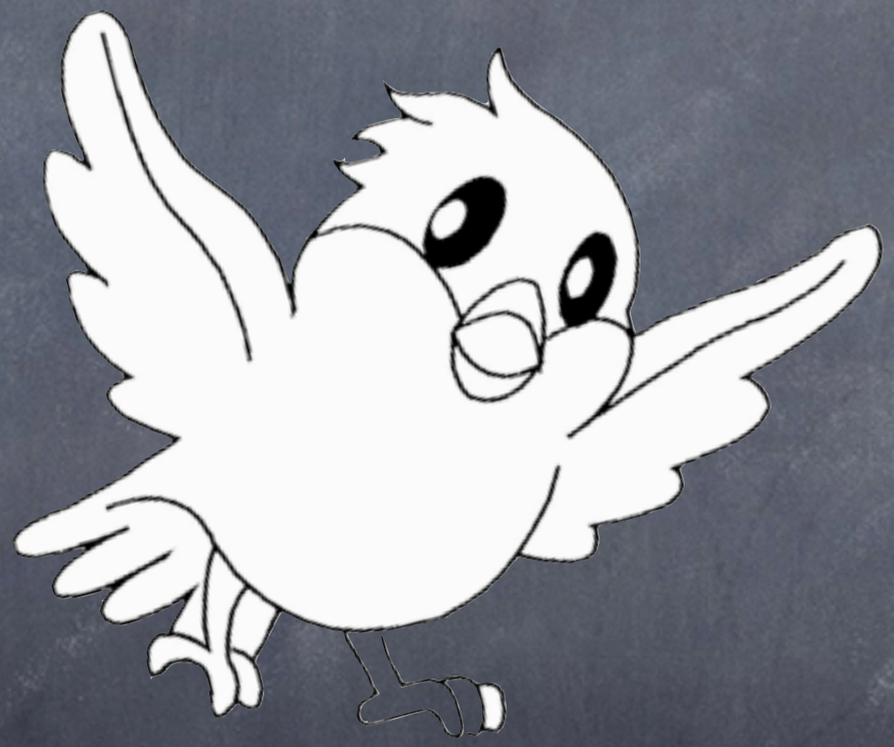
Real, large networks

- Web 2.0 Social Networks
  - Facebook, Twitter, Wikipedia, ...
- Communication Networks
  - Phone call, e-mails, ...
- And many more...
- (FB : 1 Billion Users. 1 Million new ones every day)

Temporal networks

- Traditional view: sequence of snapshots
  - A few snapshots...
- "Strongly evolving networks"
  - All modifications are considered

Multi-Agent Approach



Multi-Agent Approach

Multi-Agent Approach

Multi-Agent Approach

- Approach to deal with complex systems
- Agents with simple rules
- Environment
- Cooperation
- Emergence of global solutions from local behaviours

Multi-Agent Approach

- Strengths of MAS:
  - Adaptation to varied situations
  - Handling dynamic problems
  - Limited complexity (local behaviors)
- Difficulties with MAS
  - Find micro-level rules relevant at macro-level

Multi-Agent Approach

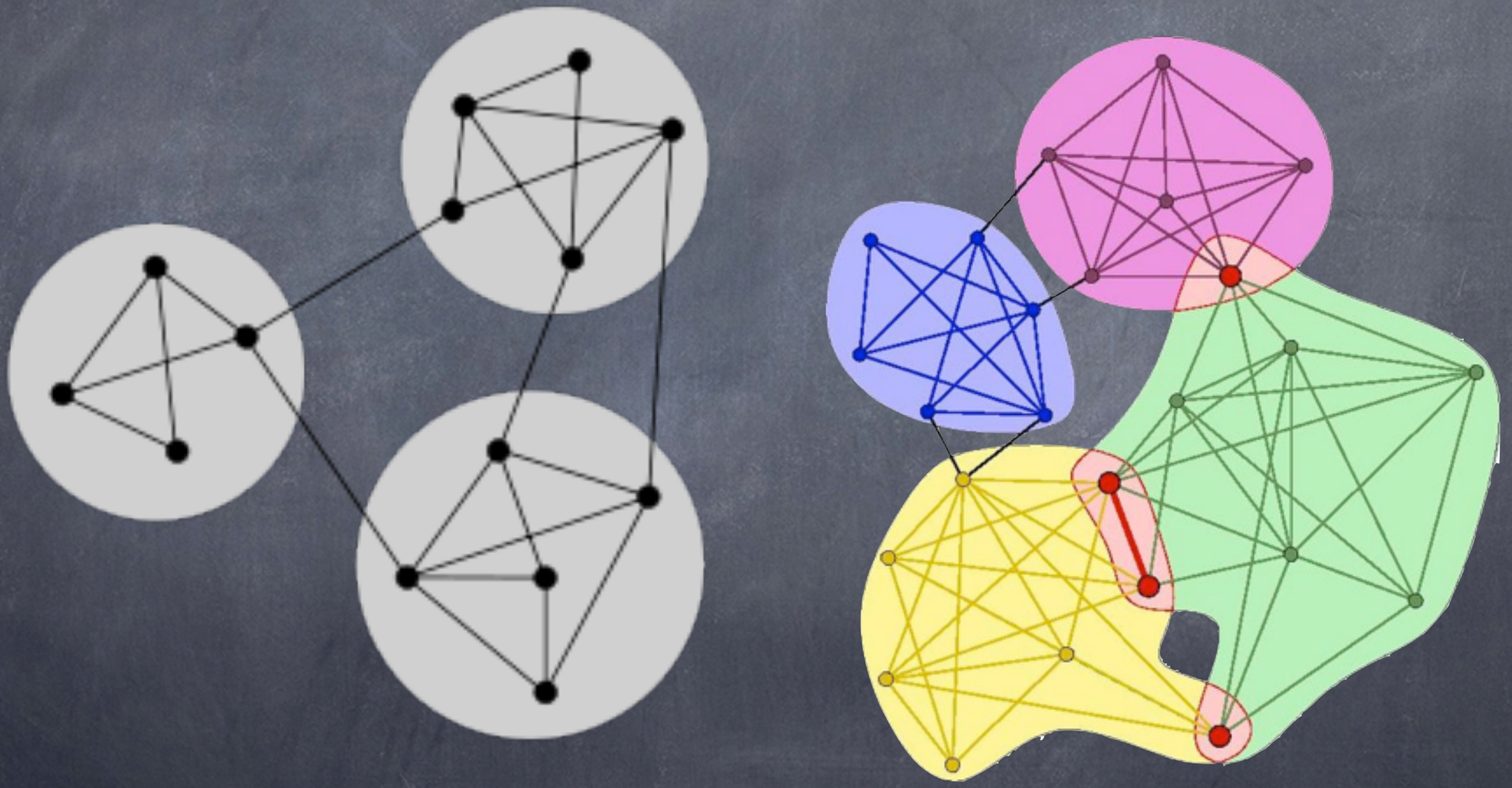
- MAS
  - Local behaviours of agents
  - Emergence of a global solution/behaviour
- Evolving Network
  - Local behaviours of nodes (attachment strategie)
  - Apparition of global phenomenon:
    - Small world
    - Power law
    - Community structure

Multi-Agent Approach

- Adaptation of MAS to community detection
  - Temporal network = evolving environment
  - Communities = agents
- Community Agents can:
  - Be born, die
  - Integrate or reject nodes
  - Be merged

Community Detection

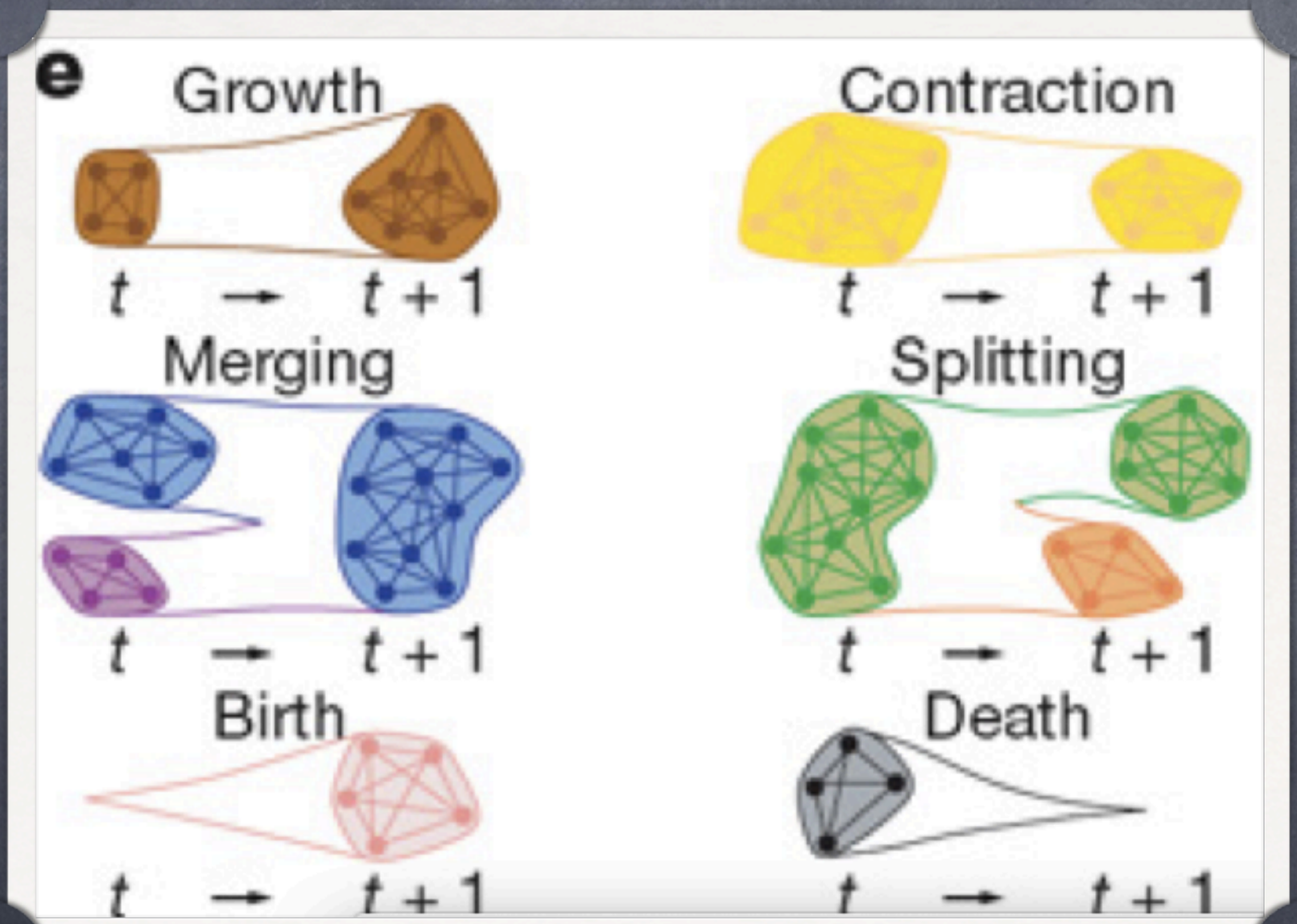
Community Detection



Community detection

- Community detection without overlap
  - GN, Louvain method, InfoMap, 200+ more...
- Community detection with overlap
  - CFinder, OSLOM, ...
- Dynamic community detection methods
  - Let's talk about them !

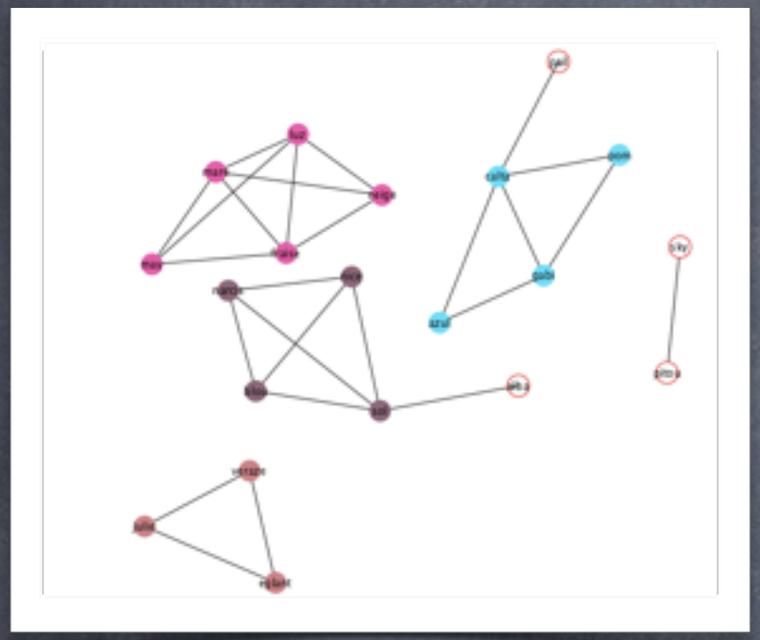
Dynamic community detection



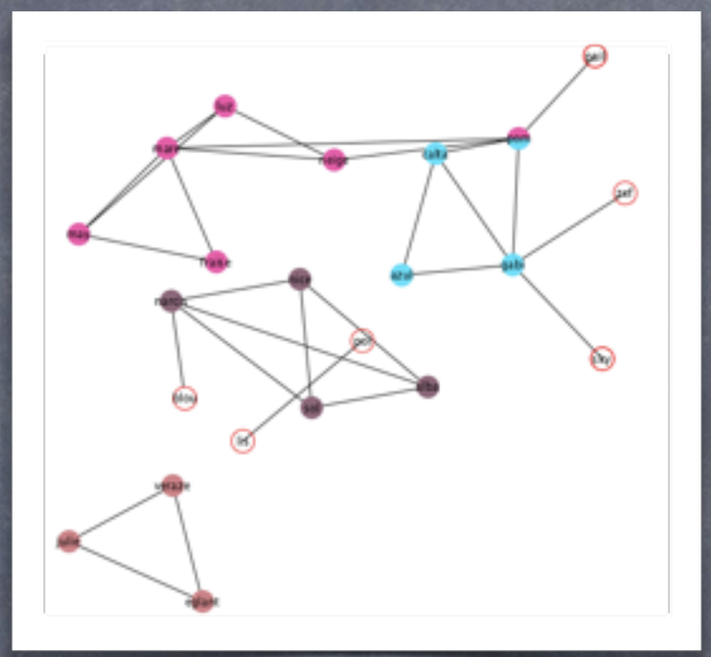
Dynamic community detection

- New problem, several approaches:
  - Independent snapshots
  - Linked snapshots
  - Temporal network

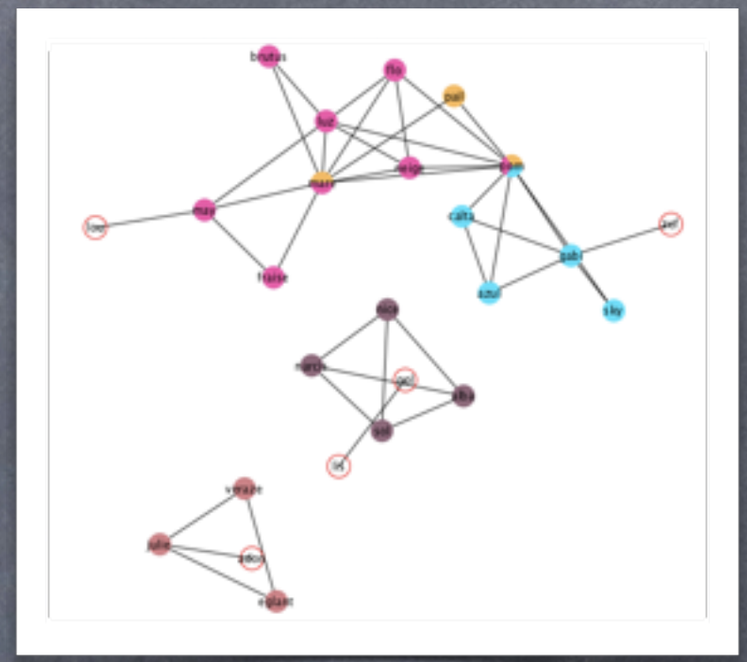
Independent snapshots



T1



T2



T3

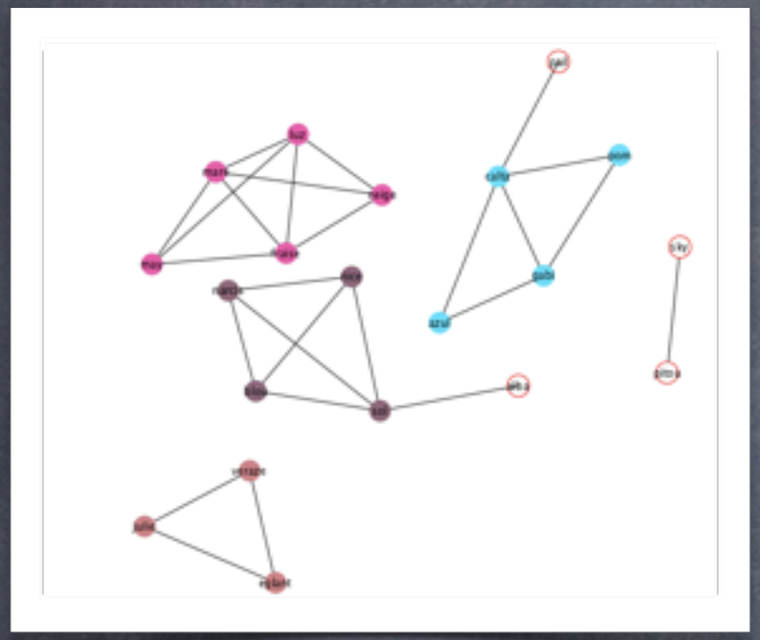
1)Detect communities on snapshots independently
2)Try to match the communities of T+1 with the ones of T

[Palla et al. 2005, Greene 2011] [Rosvall 2010(...)]

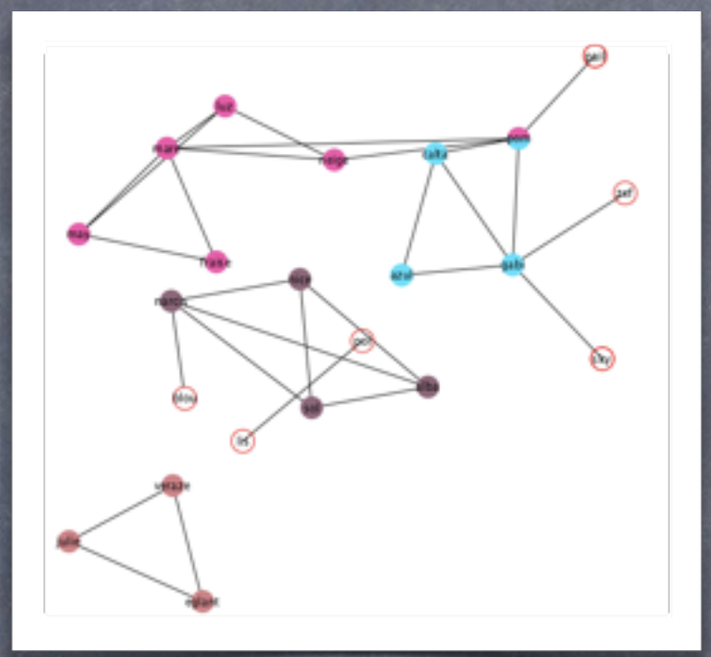
Independent snapshots

- Community detection algorithms are unstable:
  - Small variations of the network might lead to very different communities
  - Communities of T+2 close to the ones of T but not T+1

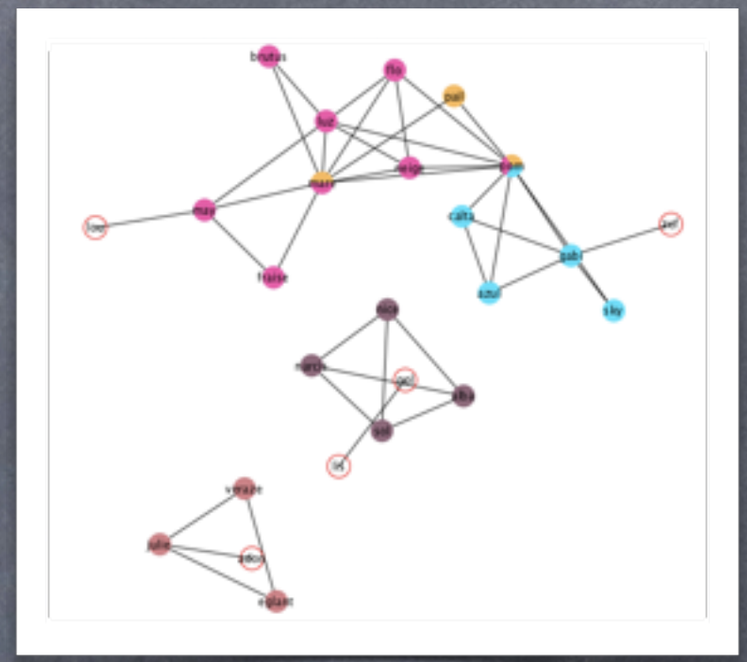
Independent snapshots



T1



T2



T3

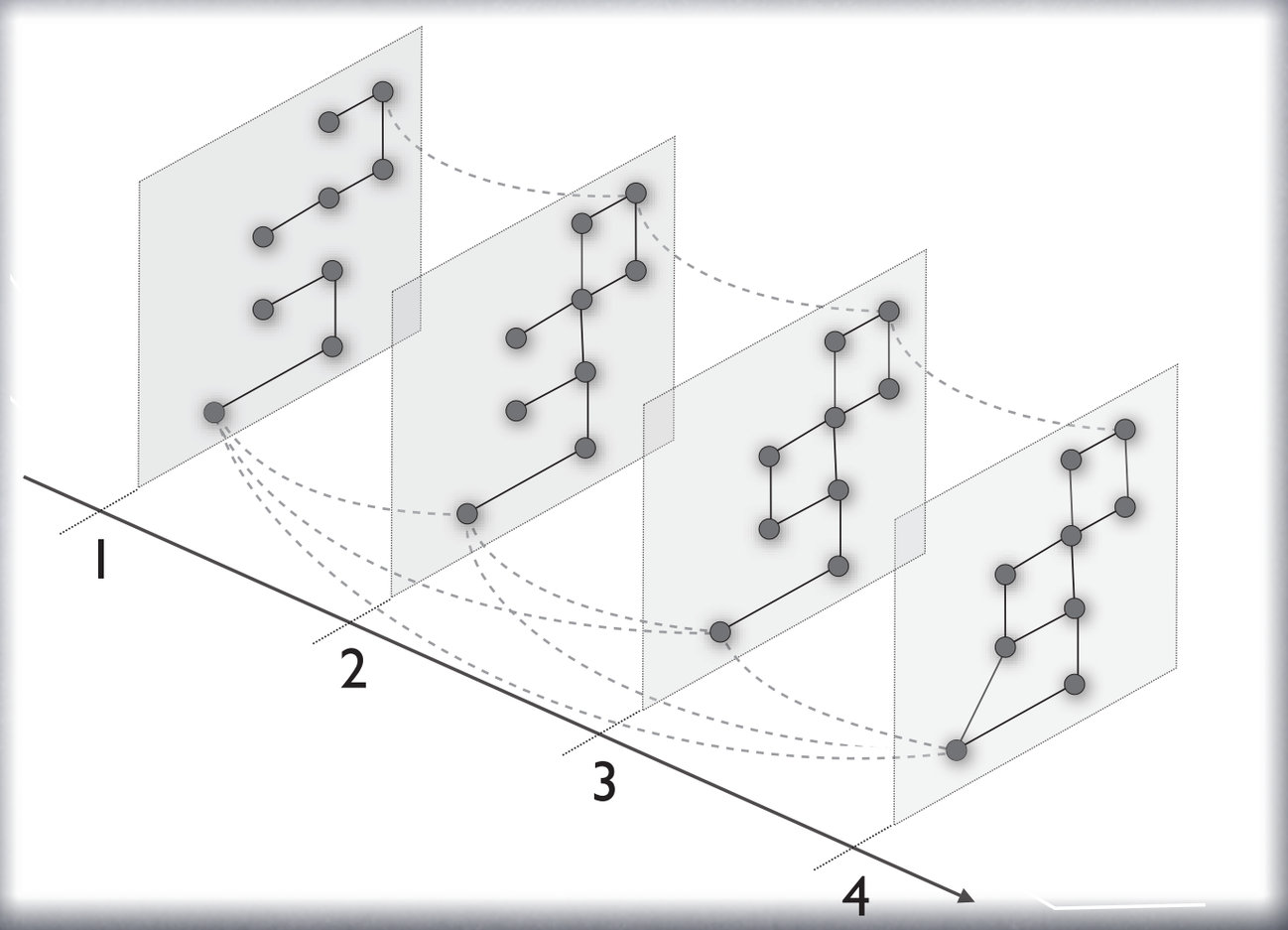
1)Detect communities on the first snapshot
2)Detect communities at T+1 by trying to get good communities while staying coherent with the previous snapshot

[Lin 2009] [Lancichinetti 2011(...)]

Linked Snapshots

1)Create a single network by linking together nodes of T which still exist at T+1

2)Run a static algorithm on this network

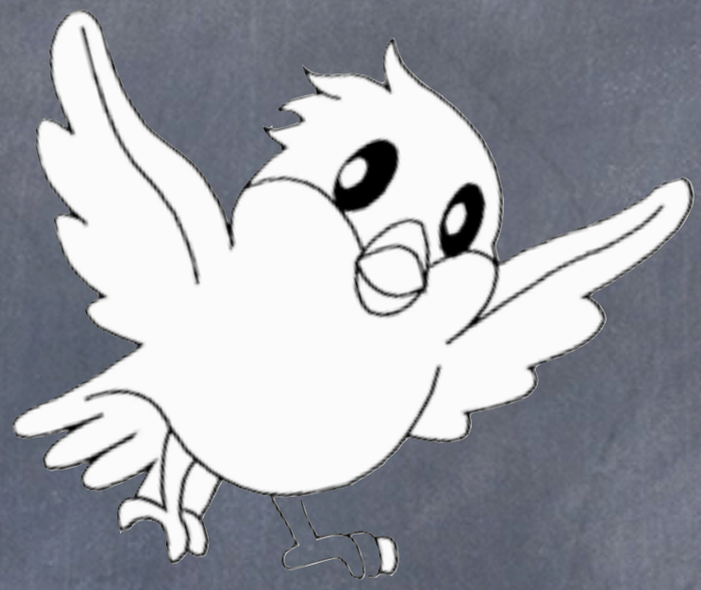


[Mucha et al. 2010, Aynaud et al. 2010]

Snapshots

- Problems with snapshots:
  - How to cut ?
    - In the middle of an event (Christmas/New year...)
    - Events happening between snapshots
  - Efficiency / Precision
    - Few snapshots: lot of variations, missing short events
    - Many snapshots: explosion of complexity

Proposed solution



Temporal network

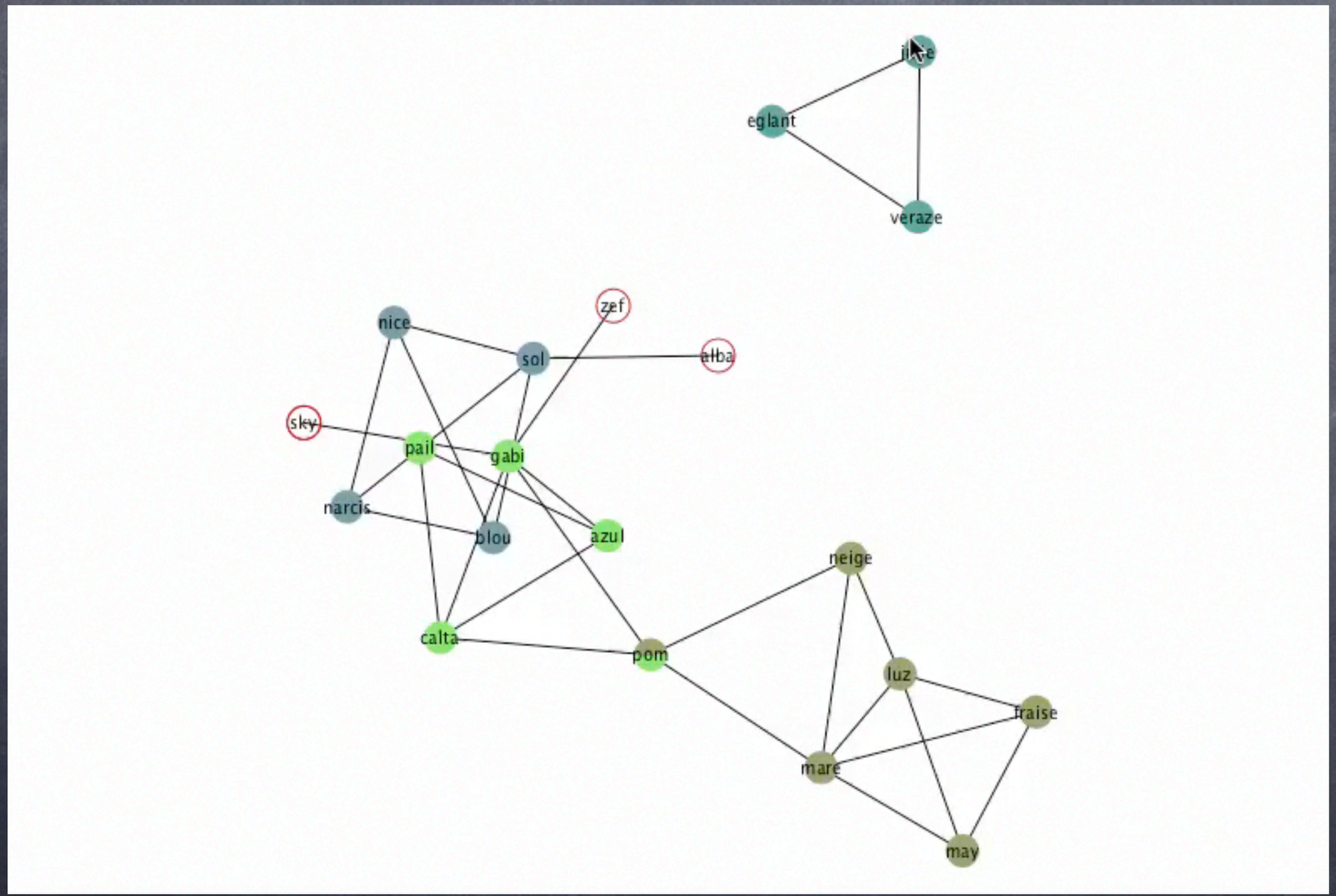
- "Interval graph" or "sequences of modification"
  - Time t: edges e1 and e2 added
  - Time t2: edges e1 and e3 removed
  - Time t3: ...
  - ...
- Real time

iLCD algorithm

- Objectives:
  - Fast (large networks, many steps)
  - Robust (noisy data, strong overlap)
  - "social like" communities (triangles, not star-like, not too sparse,...)

What does it look like

What does it look like



Algorithm

- At each step, we modify only affected communities according to local properties:
  - possibility to handle large graphs

Algorithm

- New communities are born when cliques form outside existing communities
- They die when they don't have any nodes left
- Communities integrate and reject nodes according to 3 metrics: seclusion, representativeness, potential belonging
- Communities can merge

Metrics

1) Representativeness value: The value of $representativeness(i, c)$ of a node agent i to a ommun ty c s first omputed when the agent node s added to a ommun ty Th s value s defined as

$$\frac{nbNeighb(i, c)}{k_i}$$

1) Value of seclusion: The value of $seclusion$ of the ommun ty s first omputed when the ommun ty agent s reated Th s value measures the qual ty of the ommun ty, more pre sely how well the ommun ty s separated from the rema n ng part of the system Th s value s omputed as

$$\sum_{n \in c} representativeness(n, c)$$

i c i

c i . i i

2) Decide to integrate or not a node agent: When a ommun ty agent c re e ves a request from a node agent n ask ng for h s ntegrat on, t first omputes ts potent al belong ng, $pb(c, n)$ Th s value s defined as

$$pb(c, n) = \sum_{n2 \in neighb(n)} representativeness(n2, c)$$

c i i c c i

c . i i c

ci c i i

i i . i i c

Validation

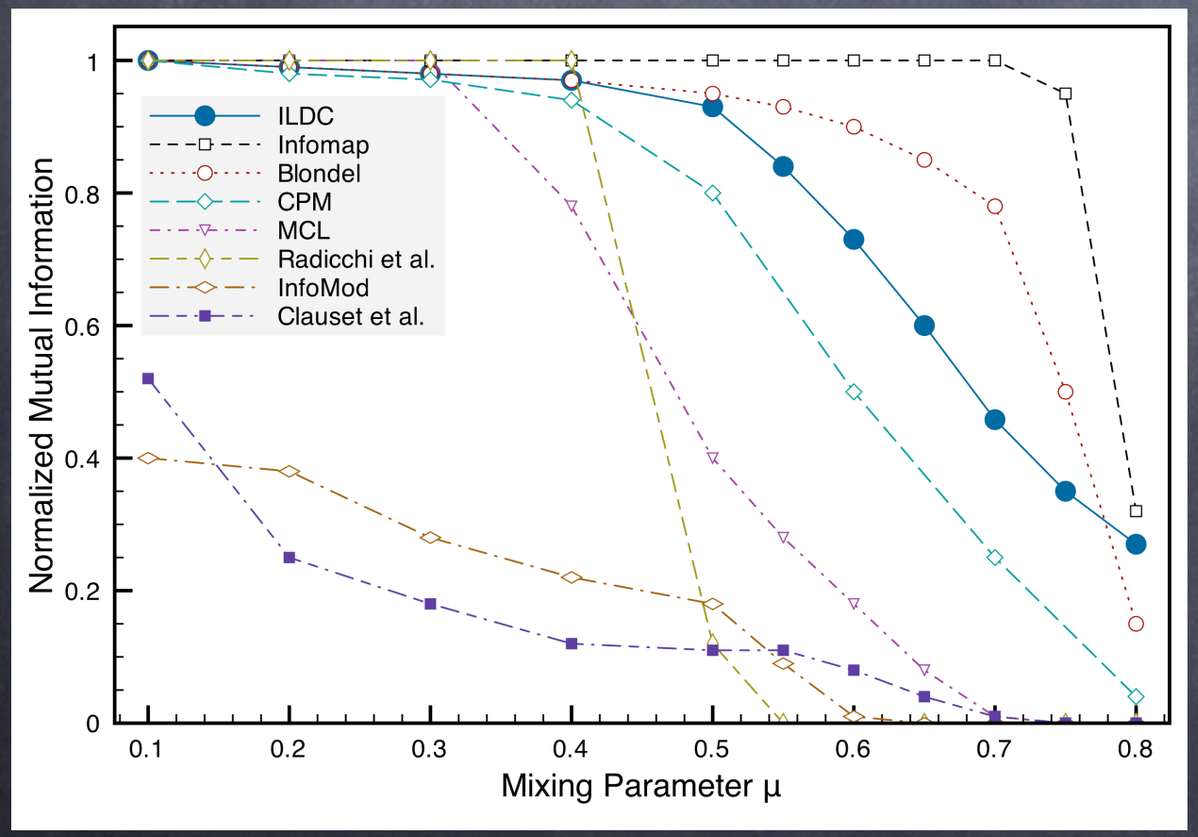
- Static point of view
  - Generated Benchmarks
  - Facebook Application
  - Comparison on real networks
- Dynamic point of view
  - Applications
  - Other algorithms ? Benchmark ? Dataset ?

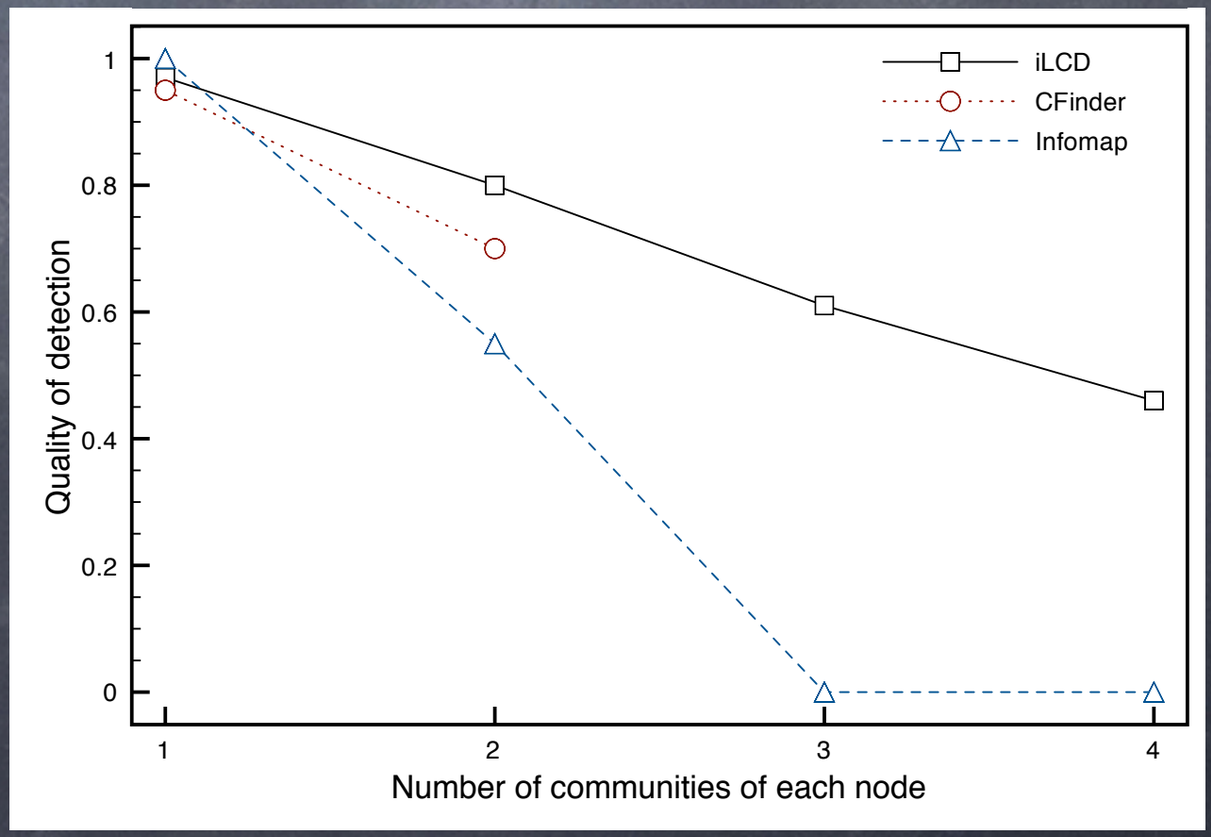
Validation : Generated benchmarks

- LFR Benchmark
  - Random order of edges
  - Unrealistic communities (too regular)
- Results:
  - Large and sparse communities = bad results
  - Dense communities = good results, even when noisy

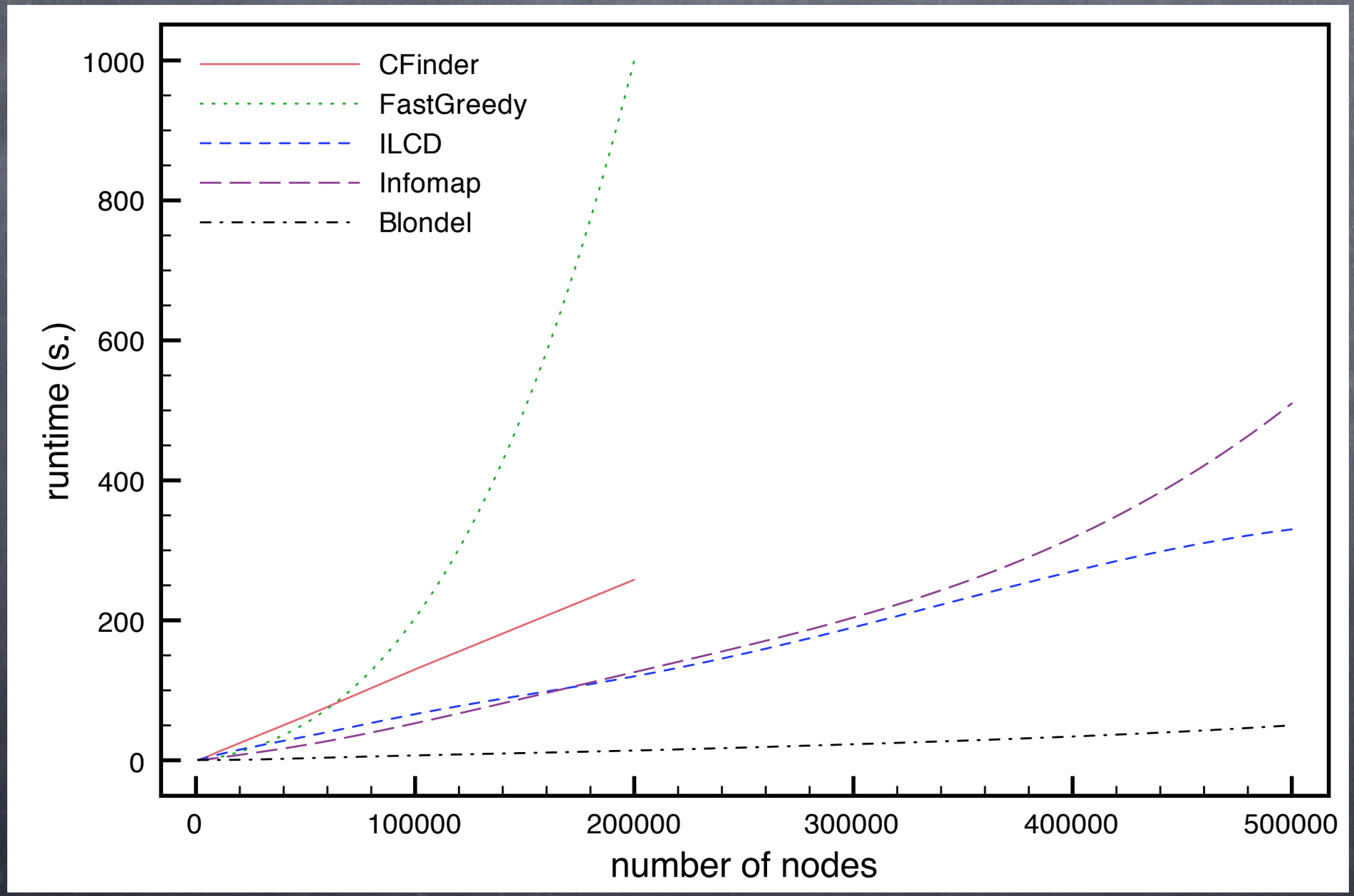
Validation : Generated benchmarks

- Networks generated to match with web 2.0 Social networks
  - Communities' size : 20-30 nodes
  - Degree of nodes : 25 * n
  - Overlap case : 25% of random edges





Validation : Speed



Validation: FB App

Validation : FB App

- Algorithms efficient on generated networks
- What about real networks ?
- Comparison of algorithms on real networks
  - "correct" communities ?

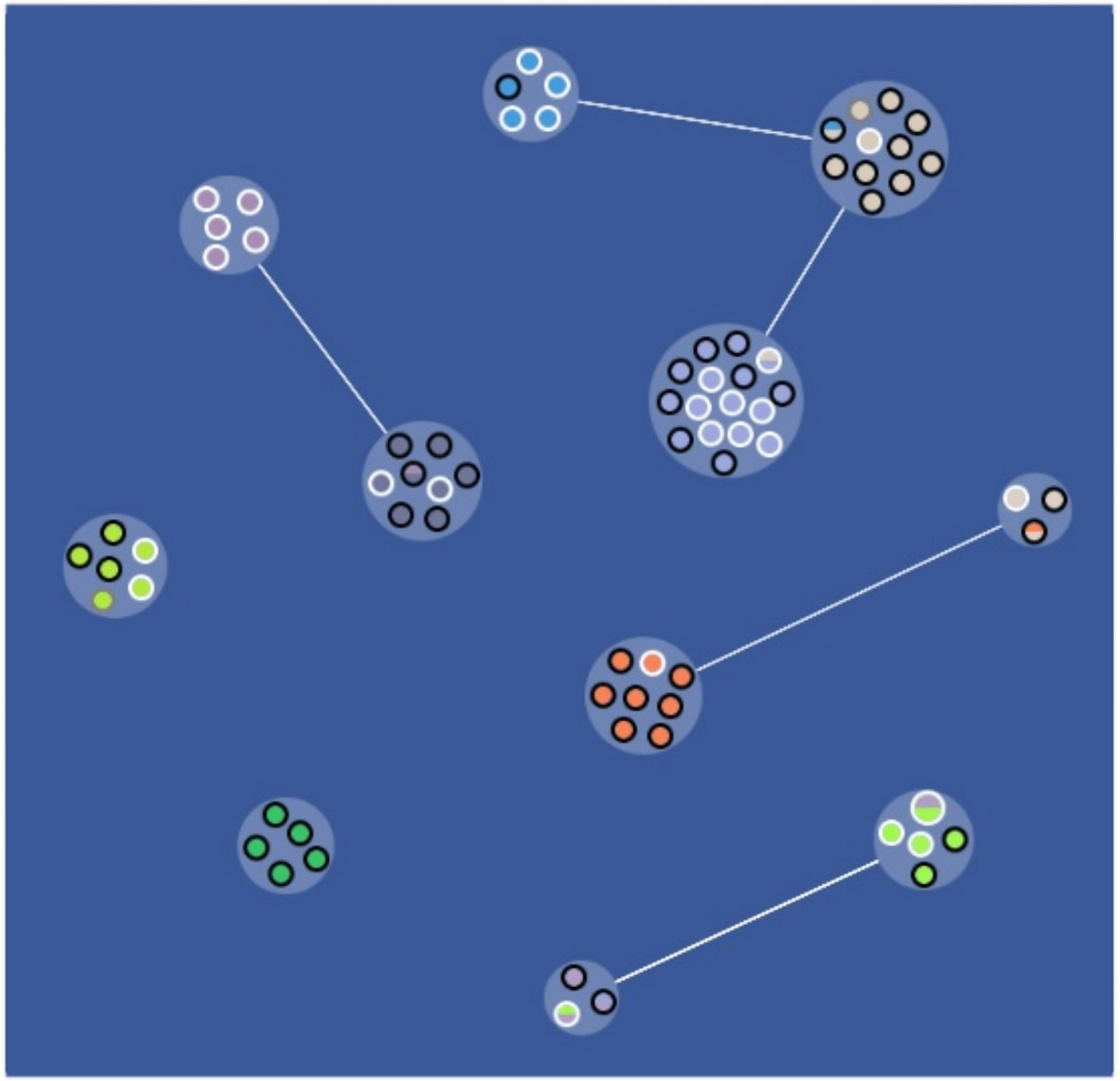
Validation : FB App

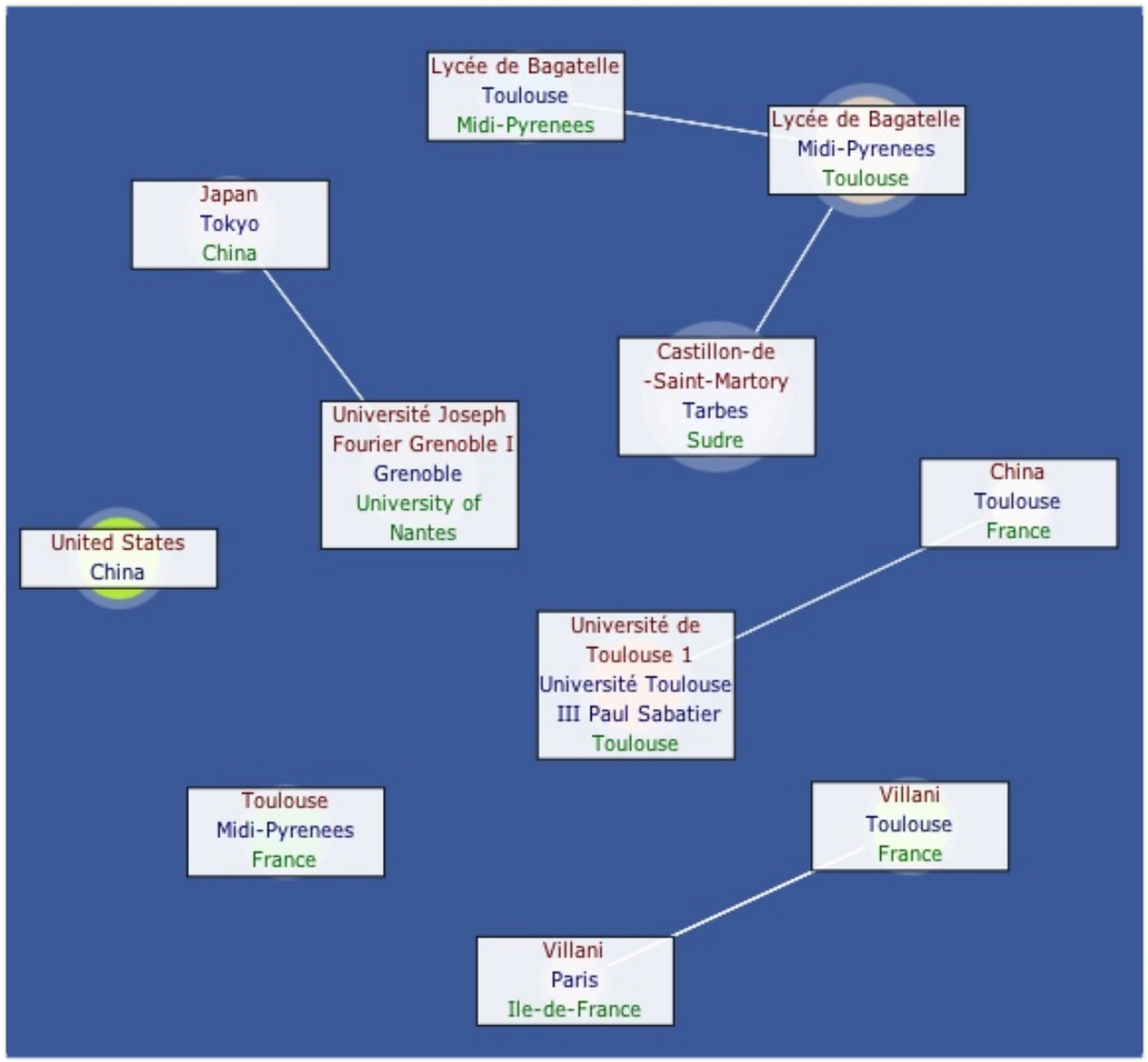
- Cooperation with sociologists
  - A need for group management on Web2.0 SN
  - Answer : automatic group detection ?
- Proposition of a solution
  - Community detection
- Development of a test application on FB

Validation : FB App

- Access ego-network of the user
- Run Community detection
- Ask user to evaluate the decomposition

Validation : FB App





Validation : FB App

- Score

  1: The solution is incoherent and/or incomprehensible

  2: The solution is bad, but there are some good things

  3: Average Solution

  4: The solution is good and logical, but there are a few mistakes

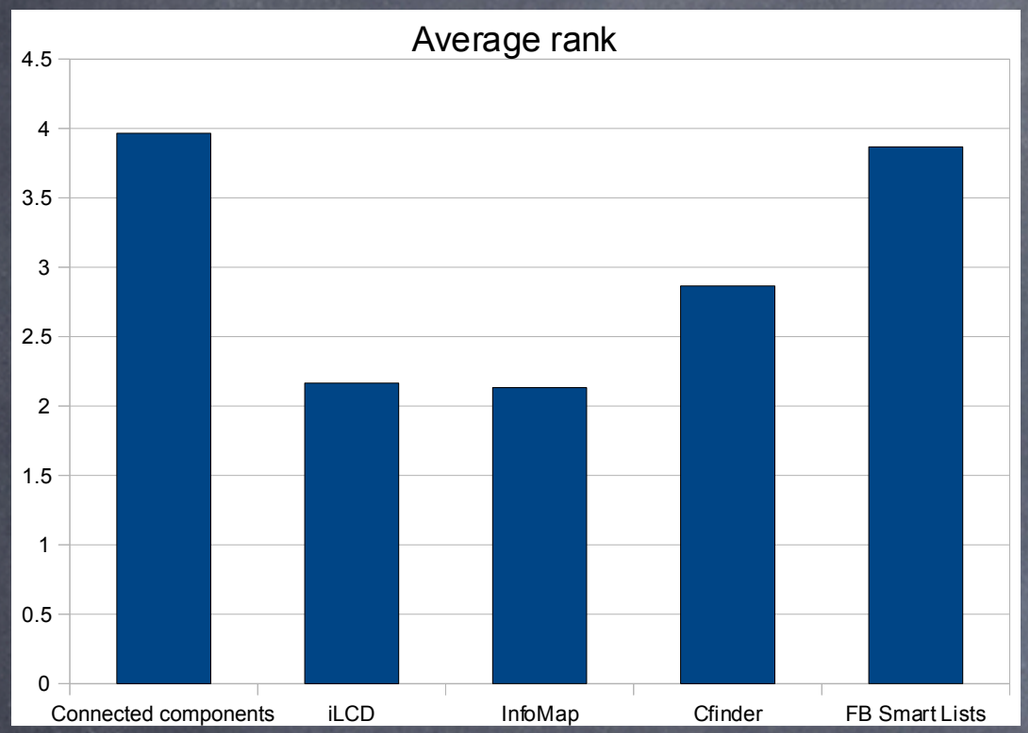
  5: The solution is perfect

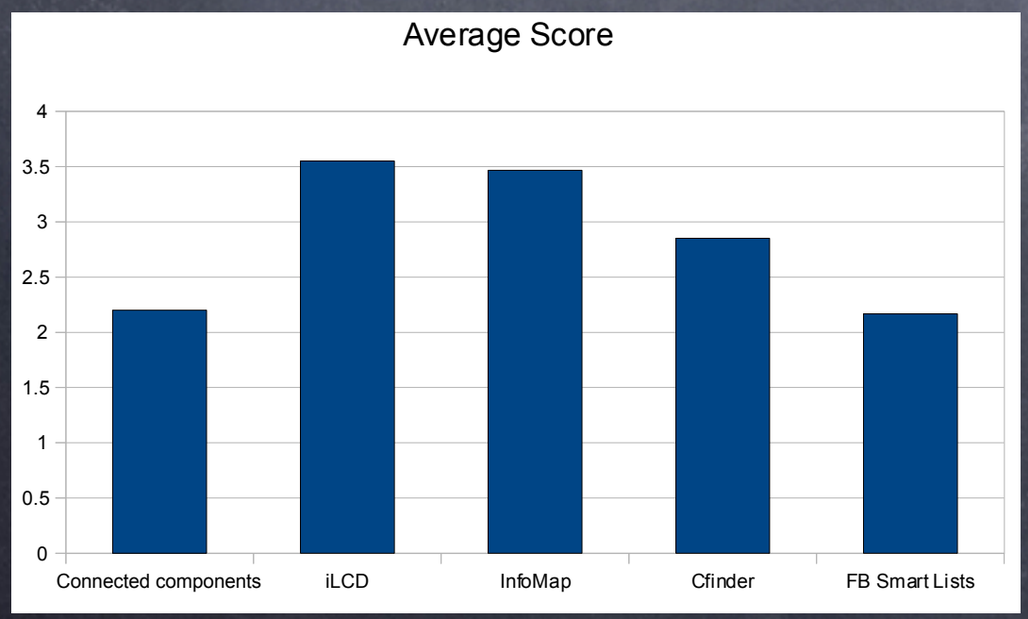
- Ranking
  - 1-5, tie accepted

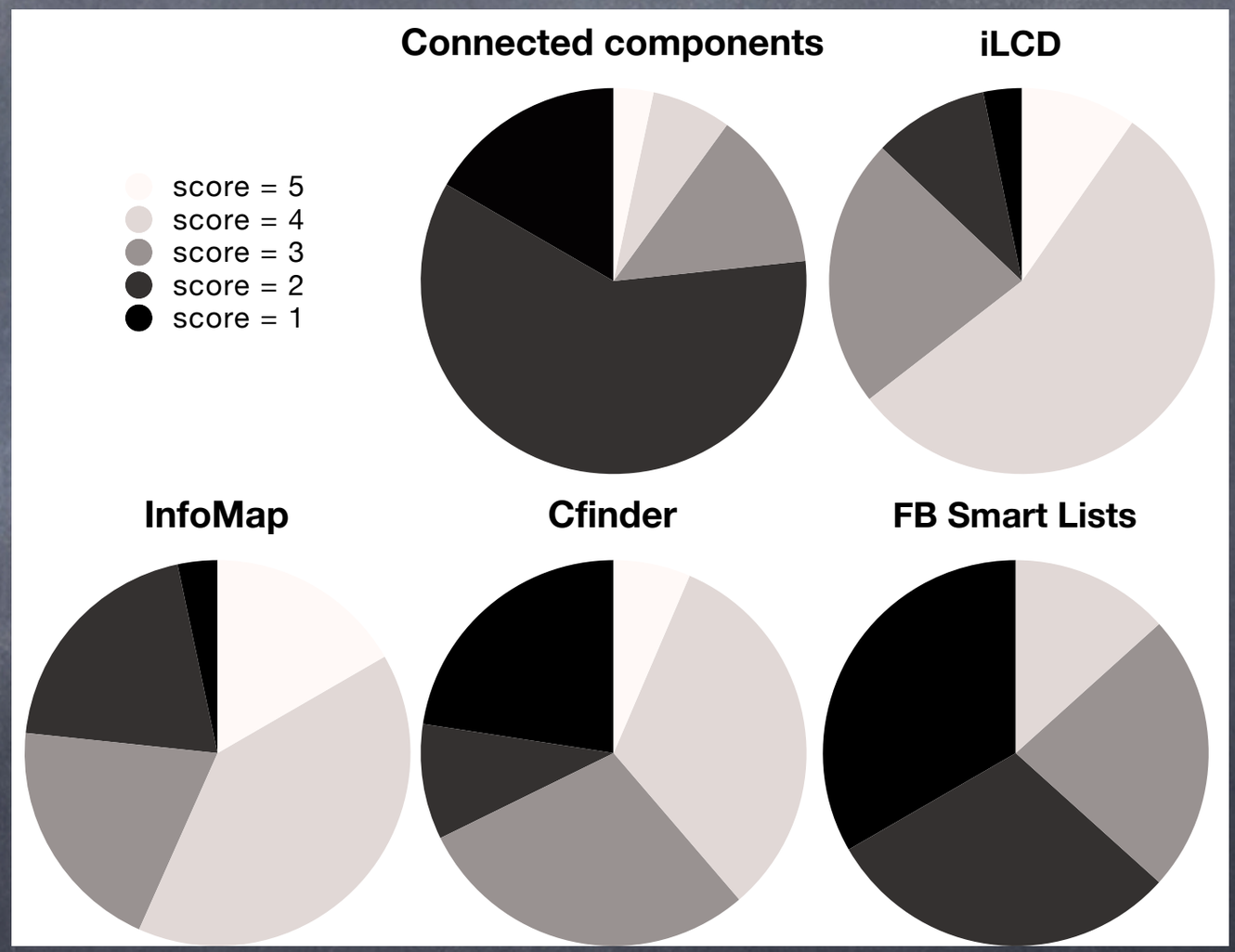
Validation : FB App

- Tested solutions:
  - iLCD
  - InfoMap
  - CFinder
  - Connected components
  - FB Smart Lists

Validation : FB App

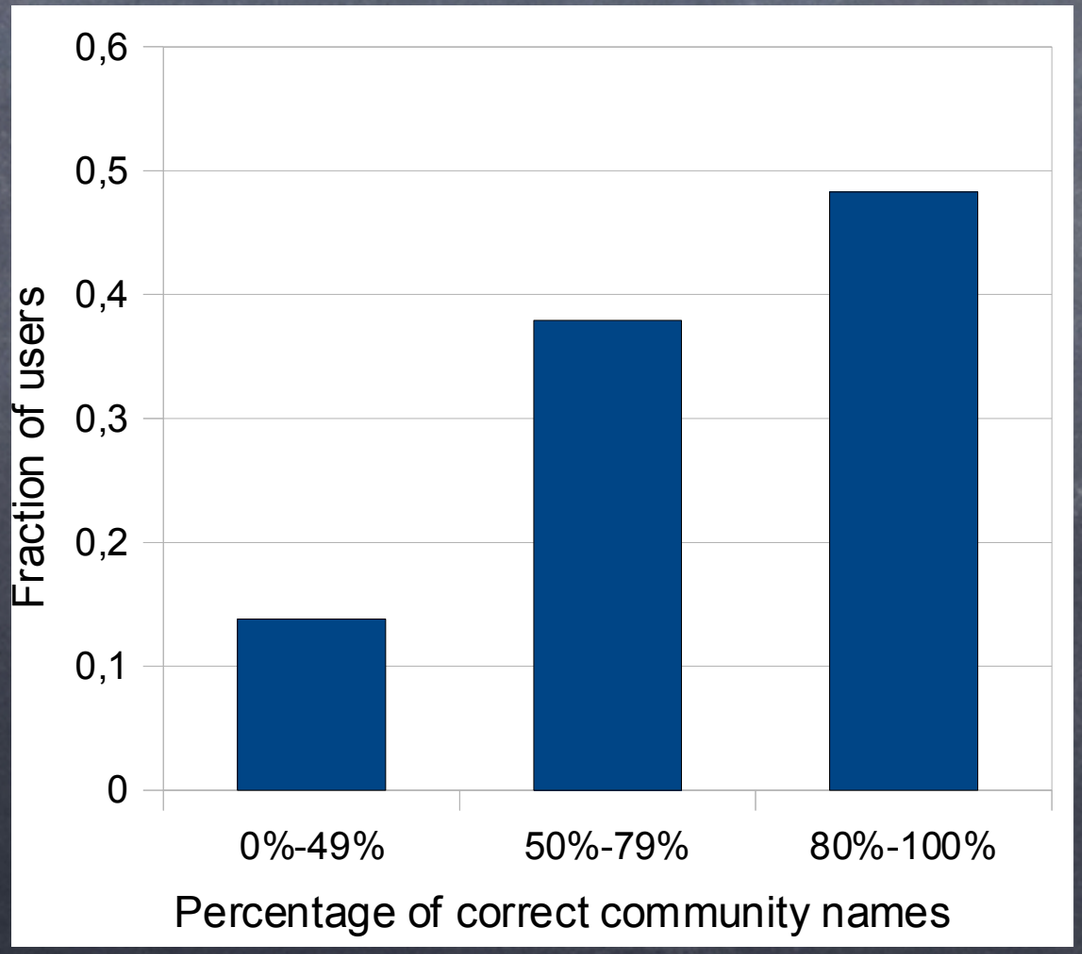






FB App: what else ?

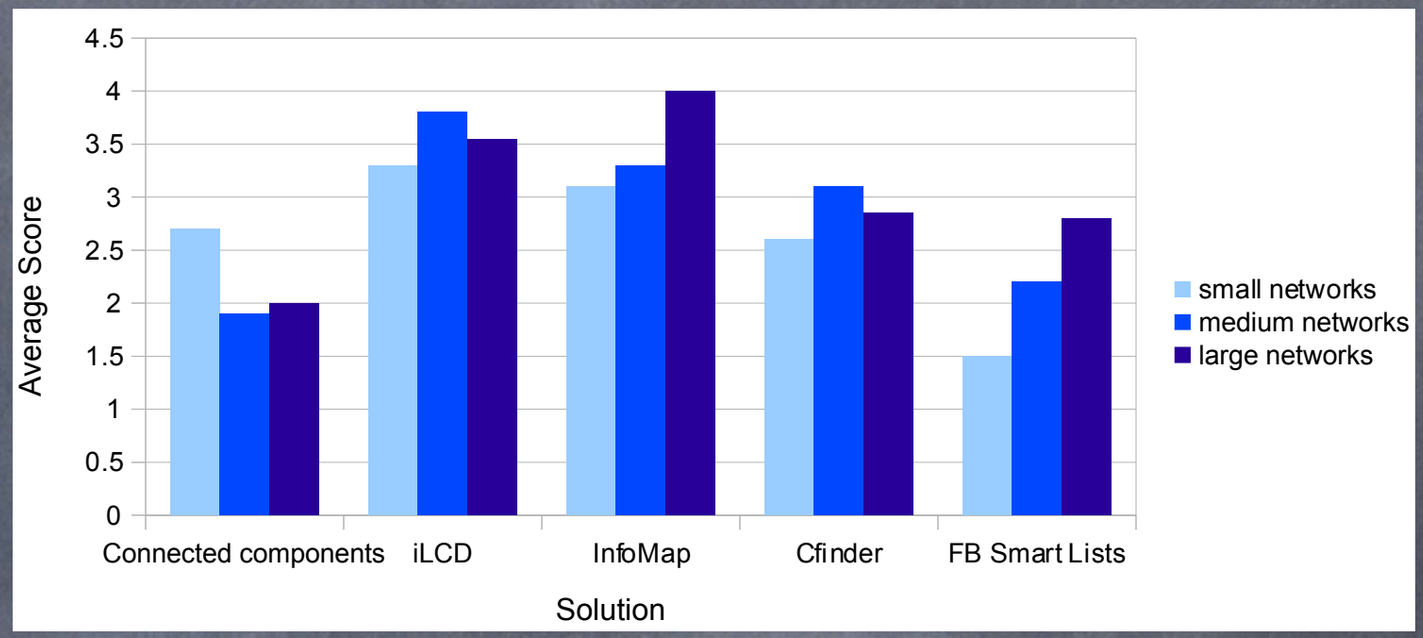
- Community naming:
  - ratio term frequency in community / TF in whole network
  - 3 most significant terms



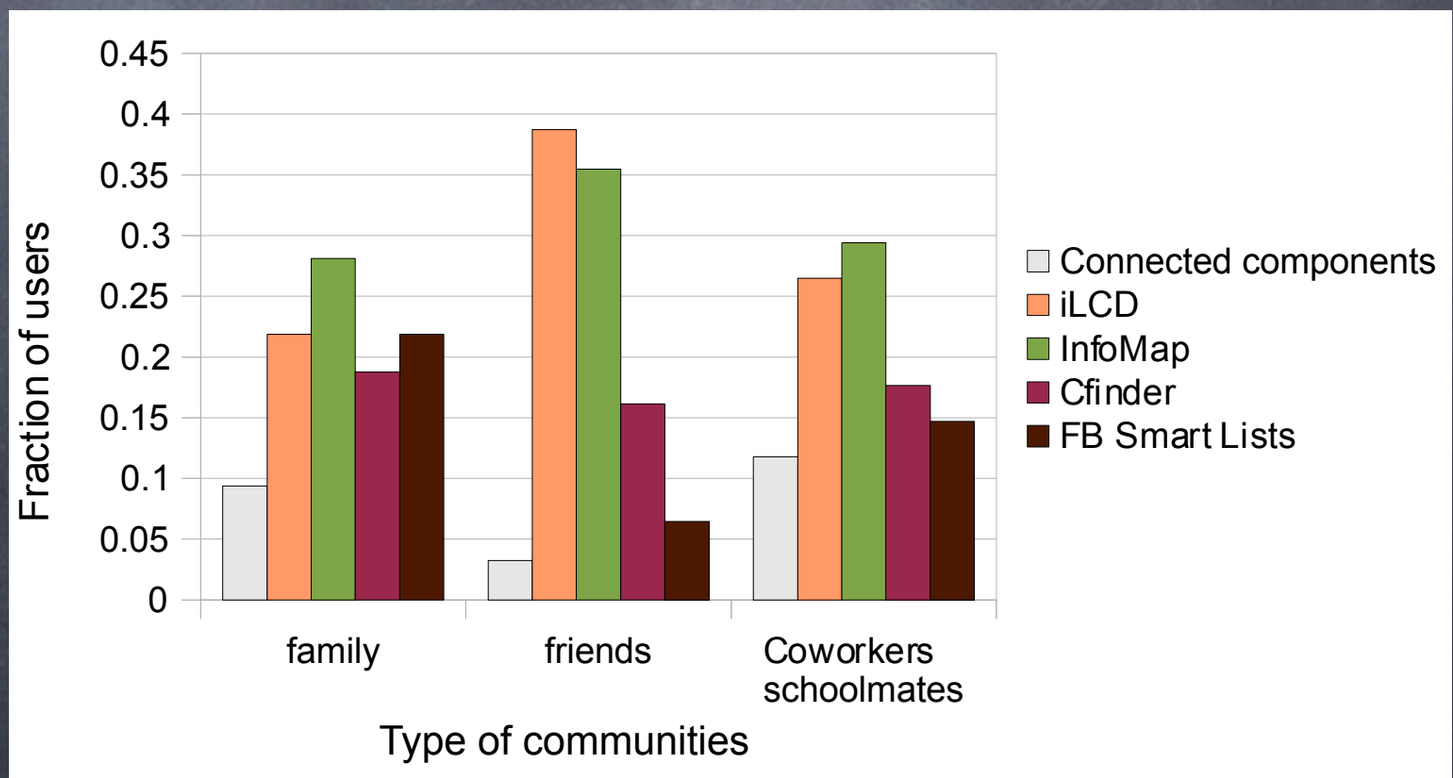
FB App: what else ?

Efficiency according to network properties:

Size



Community type



FB App: conclusions

- Only 30 users...
- iLCD: efficient
- Community detection: efficient
- Users are afraid...
- ...But think it's useful

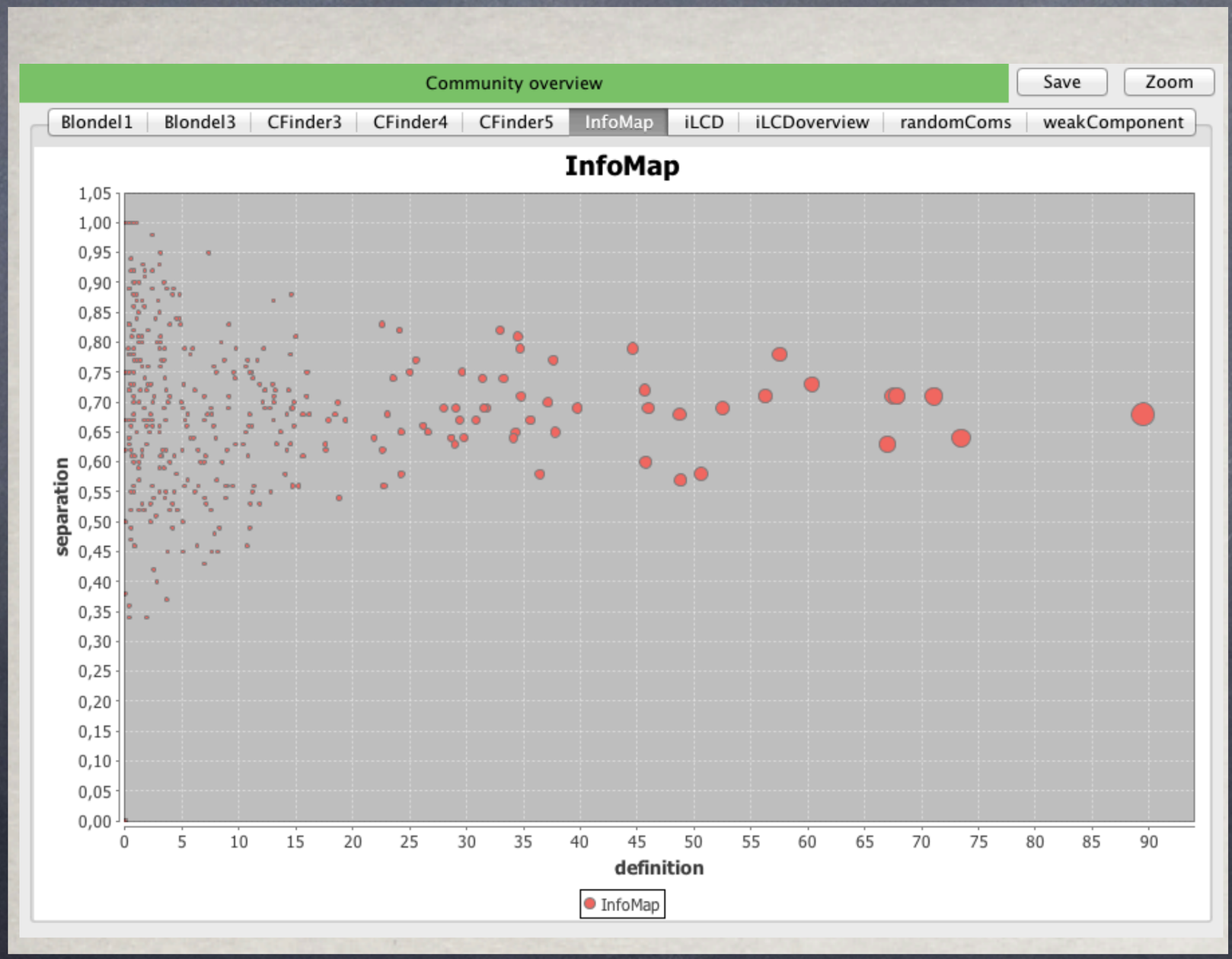
Comparing communities on real networks

- Different algorithms = different results
  - Perfect algorithm efficient in all cases ?
  - Several pertinent solutions ?
  - To each network its best solution ?
- How to know if a solution is pertinent ?
  - Evaluation by expert (but...)
  - Tool to explore & compare solutions

Circle's circumference = Community size

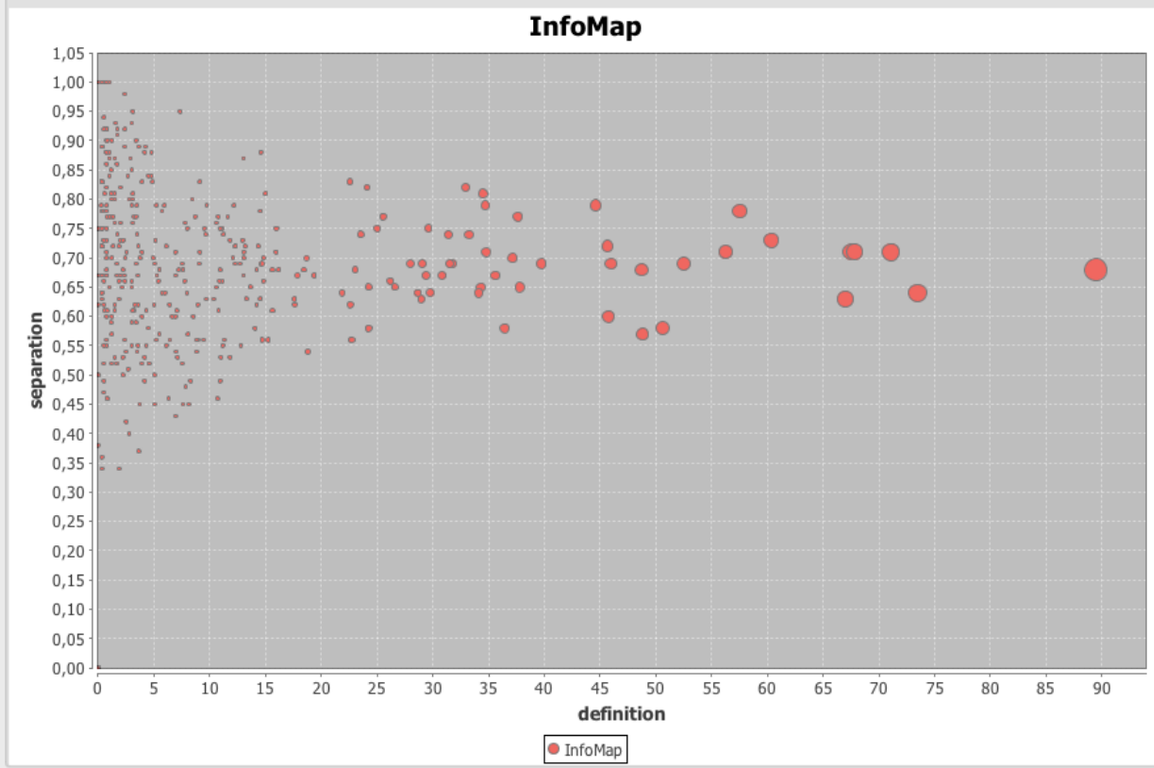
Separation = extern edges/nb edges

Definition = nb possible edges lacking/nbNodes



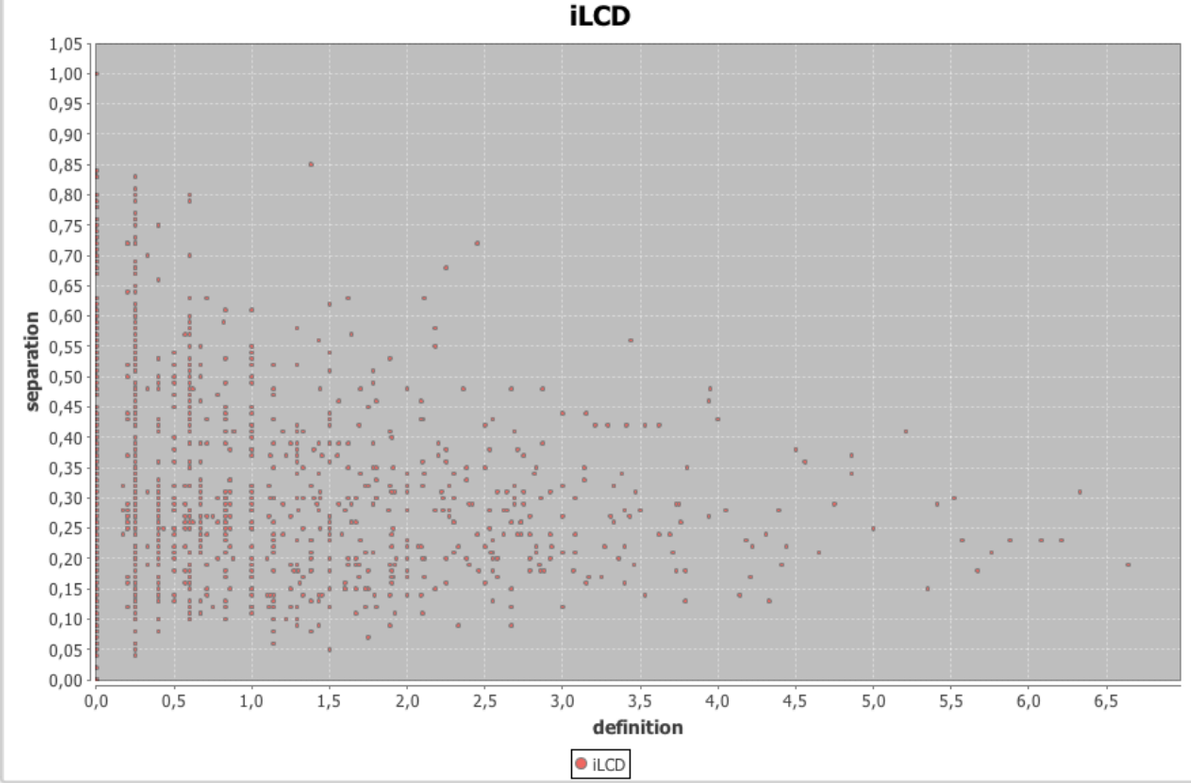
Community overview | Save | Zoom

Blondel1 | Blondel3 | CFinder3 | CFinder4 | CFinder5 | InfoMap | iLCD | iLCDoverview | randomComs | weakComponent



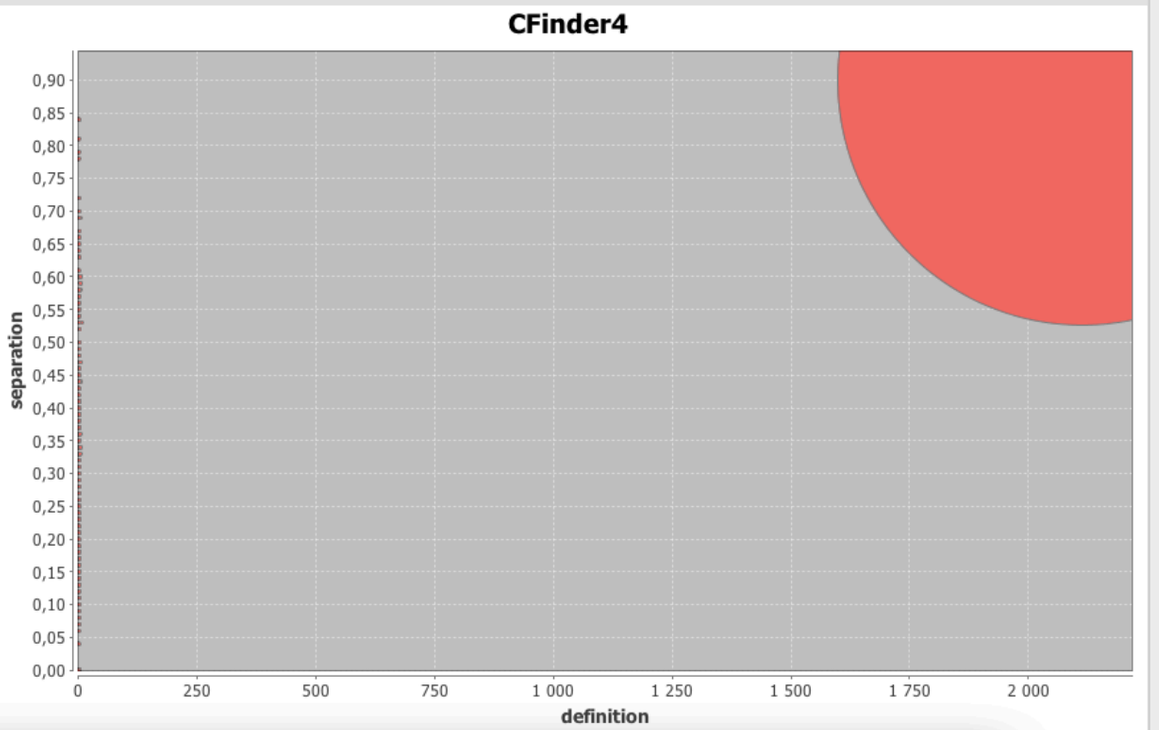
Community overview | Save | Zoom

Blondel1 | Blondel3 | CFinder3 | CFinder4 | CFinder5 | InfoMap | iLCD | iLCDoverview | randomComs | weakComponent



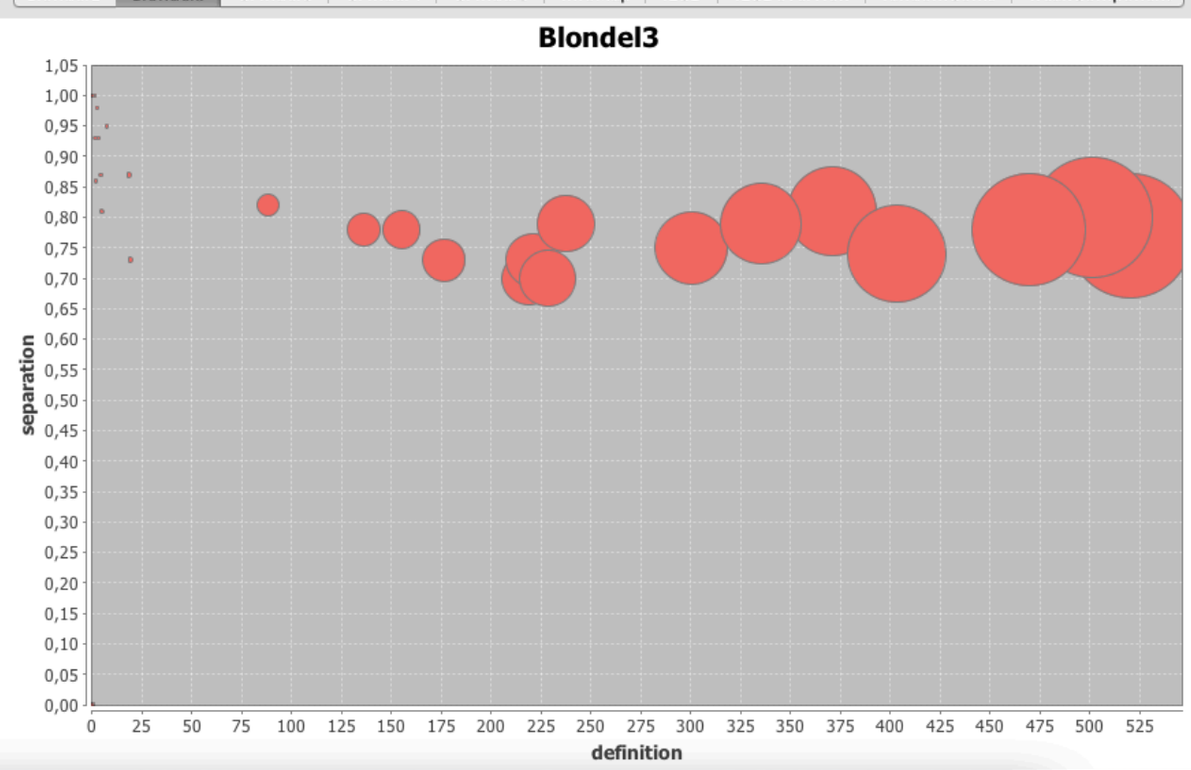
Community overview | Save | Zoom

Blondel1 | Blondel3 | CFinder3 | CFinder4 | CFinder5 | InfoMap | iLCD | iLCDoverview | randomComs | weakComponent



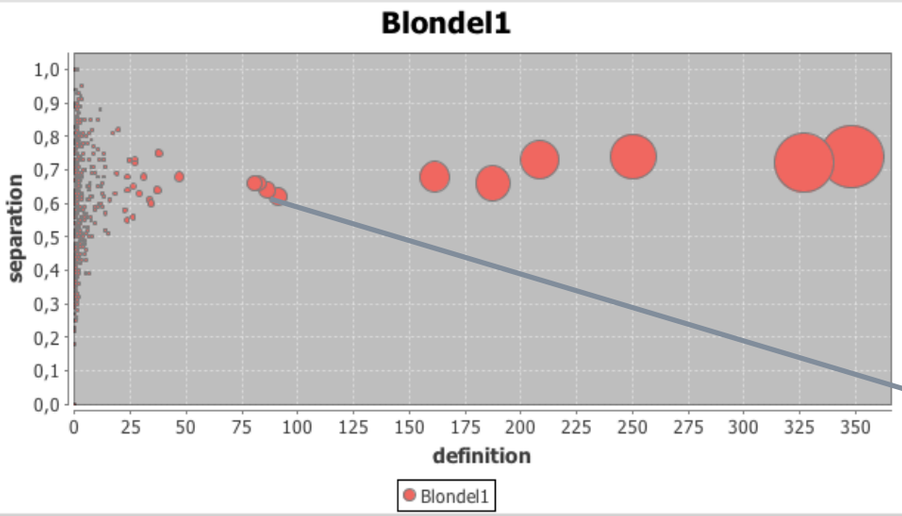
Community overview | Save | Zoom

Blondel1 | Blondel3 | CFinder3 | CFinder4 | CFinder5 | InfoMap | iLCD | iLCDoverview | randomComs | weakComponent



Community overview | Save | Zoom

Blondel1 | Blondel3 | CFinder3 | CFinder4 | CFinder5 | InfoMap | iLCD | ▶



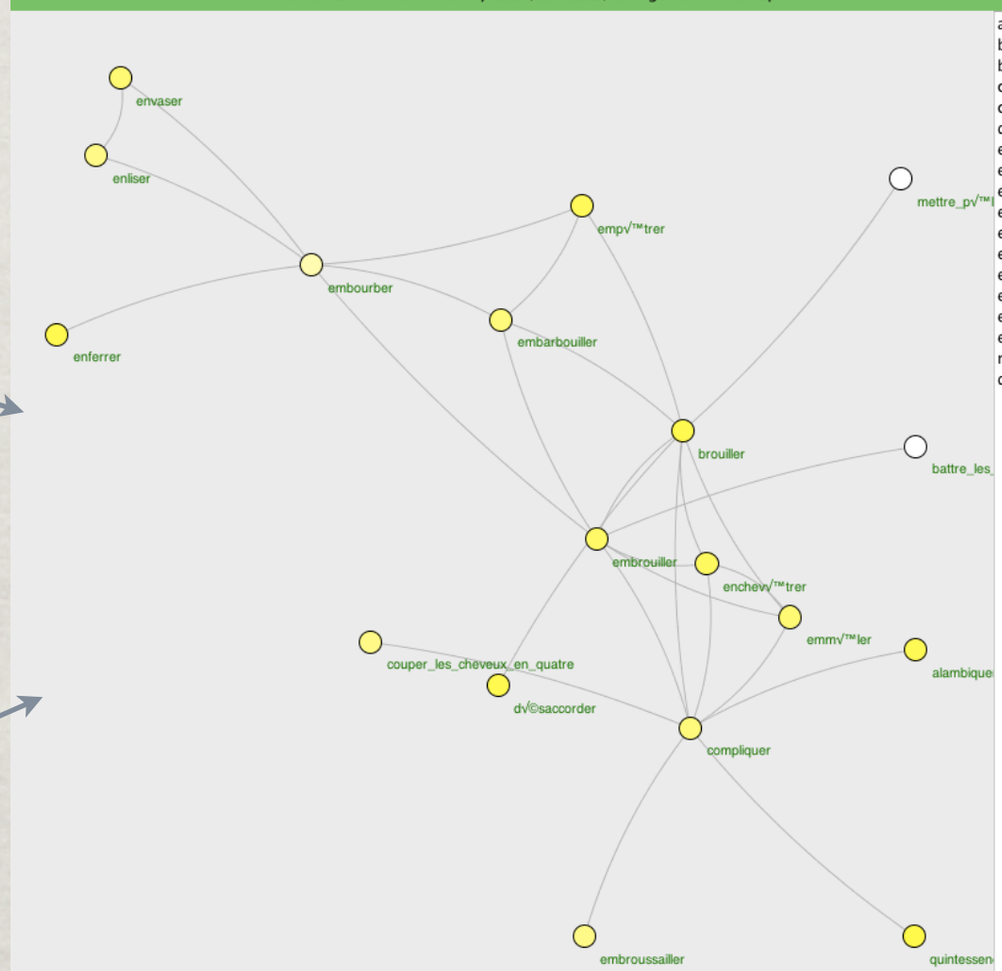
Browse nodes/communities | Save | Zoom

choose an algorithm: Blondel1

browse by node - com... ☑: abaisser — visualize

browse by community: 1477 (707 nodes) — visualize

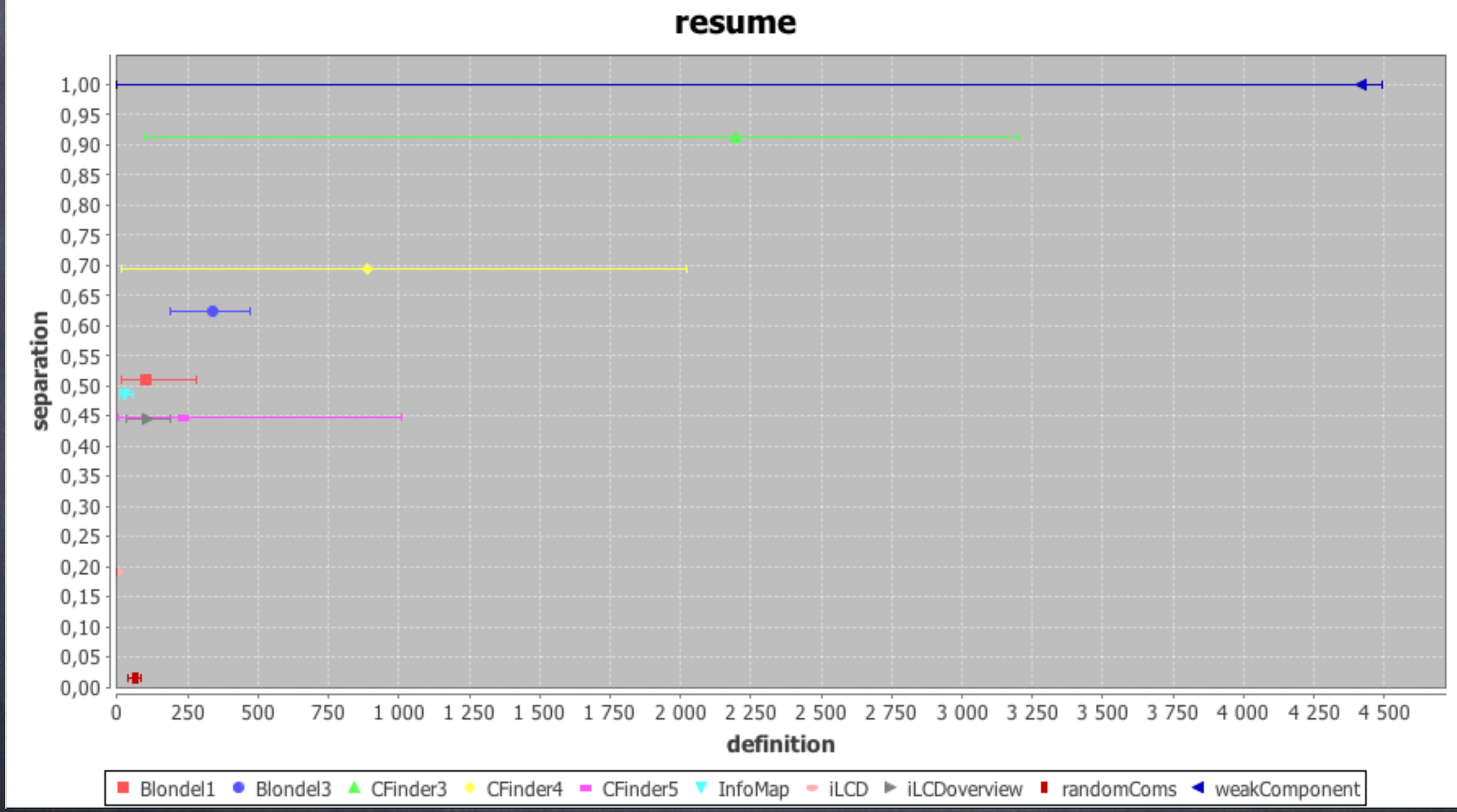
Visualisation of community 205 (18 nodes) of algorithm InfoMap | Save | Zoom



alambiquer
battre_les_cartes
brouiller
compliquer
couper_les_cheveux_en_quatre
d√©saccorder
embarbouiller
embourber
embrouiller
embroussailler
emm√™ler
emp√™trer
enchev√™trer
enferrer
enliser
envaser
mettre_p√™le-m√™le
quintessencier

2 dimension community evaluation

Save Zoom

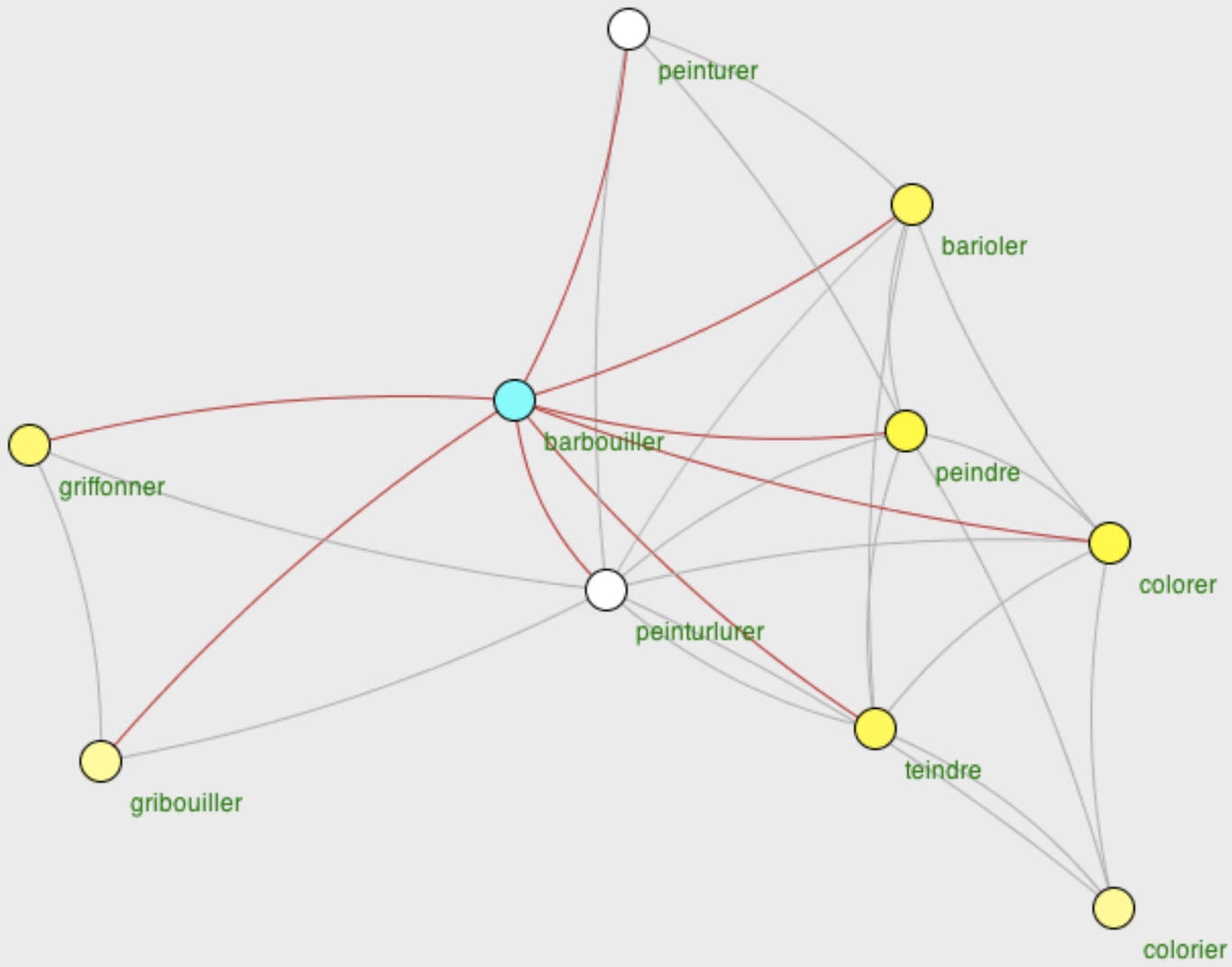


Communities explorer

- Possible problems
  - Detection of super-communities (Navarro - Cazabet)
  - Unclassified nodes
  - Communities too small/bigs

Visualisation of community 2628 (10 nodes) of algorithm iLCD

Save Zoom



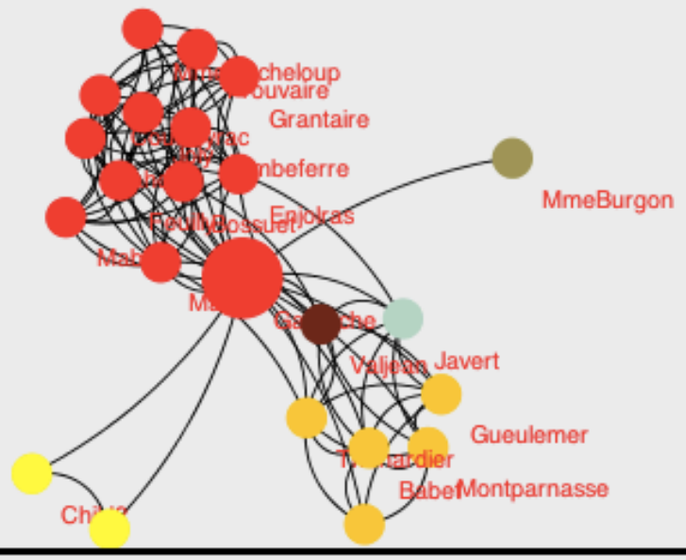
barbouiller
barioler
colorer
colorier
gribouiller
griffonner
peindre
peinturer
peinturlurer
teindre

Neighborood of Gavroche with Blondel1

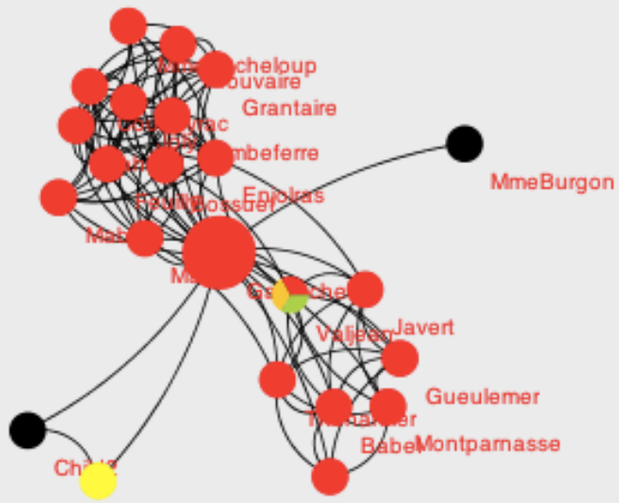
Save

Zoom

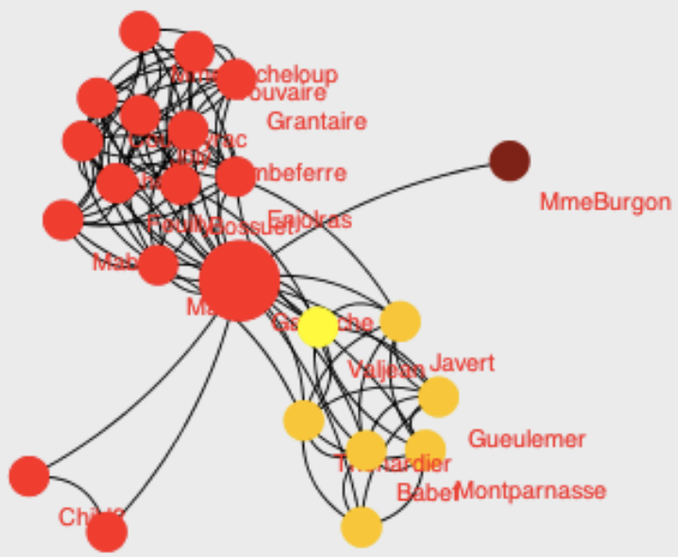
Blondel1



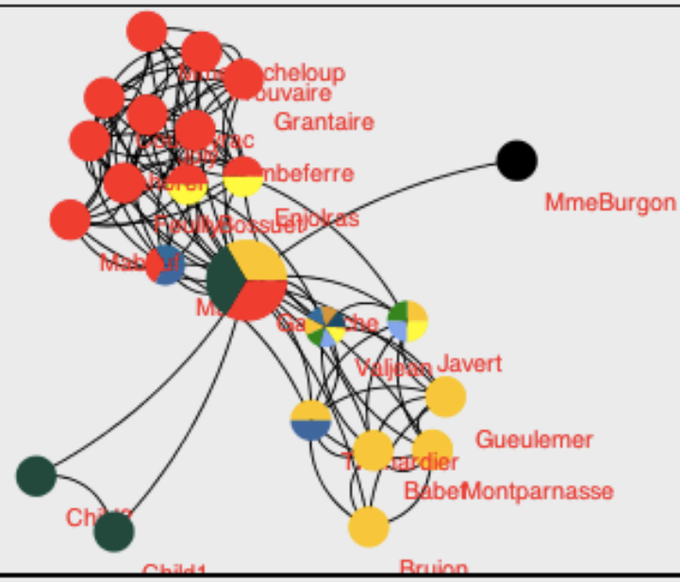
CFinder4



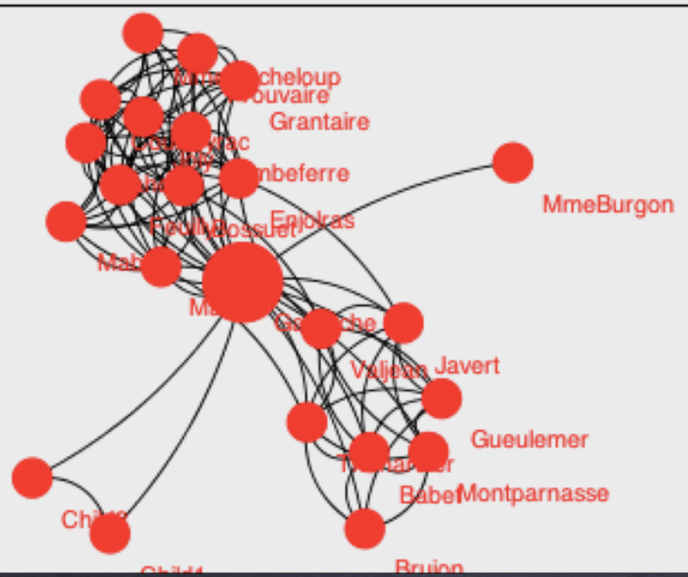
InfoMap



iLCD



weakComponent

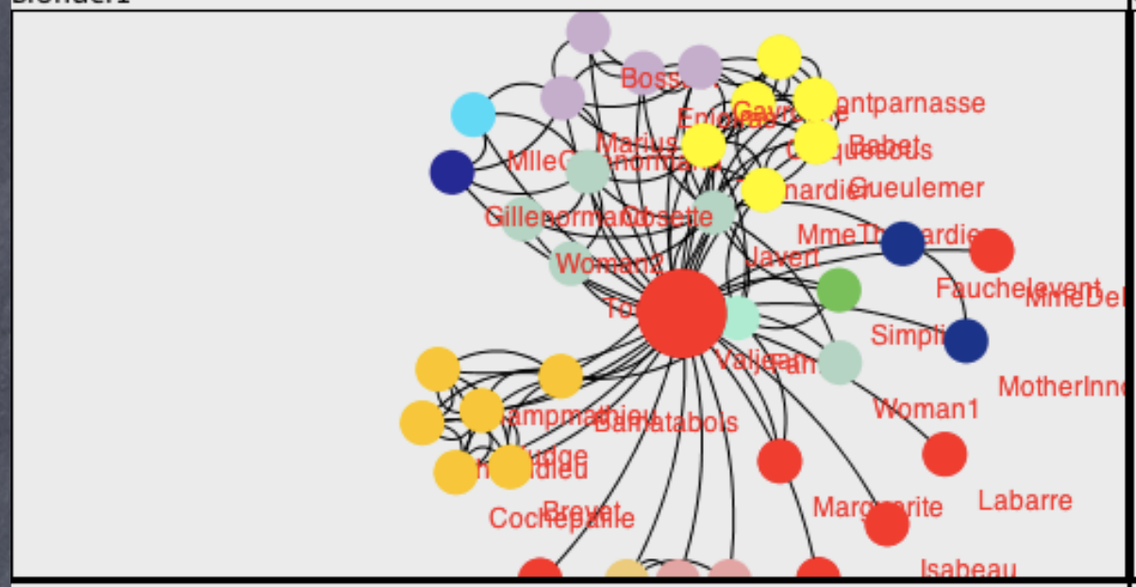


Neighborood of Valjean with Blondel1

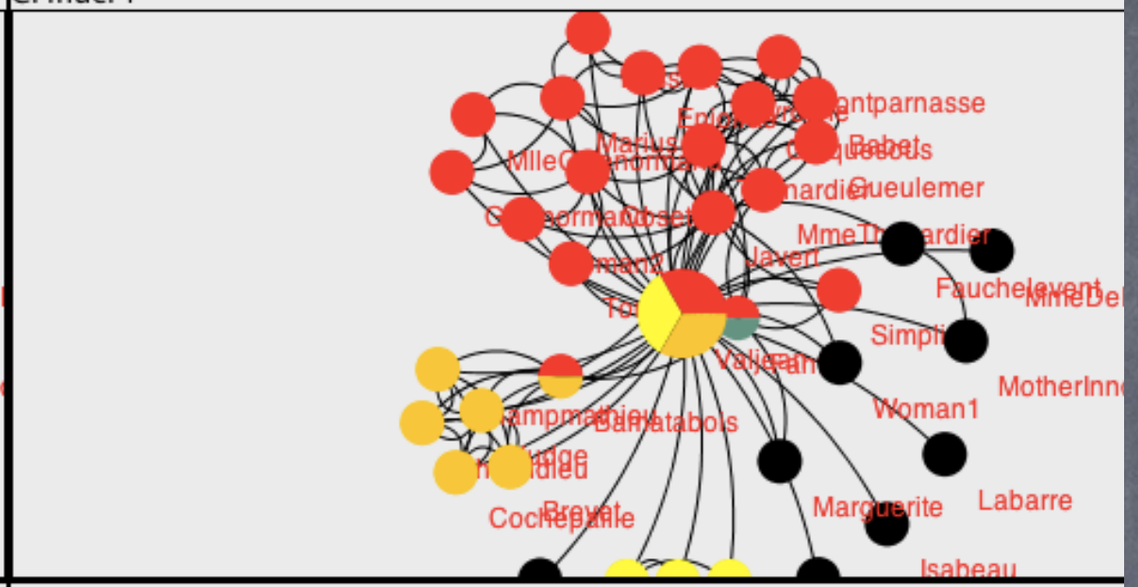
Save

Zoom

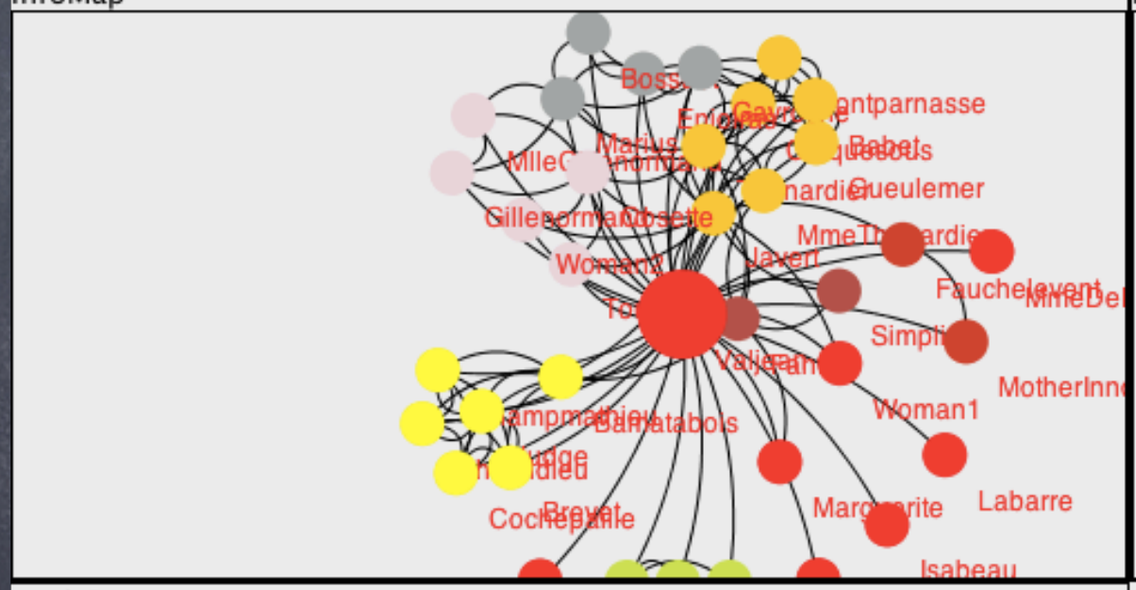
Blondel1



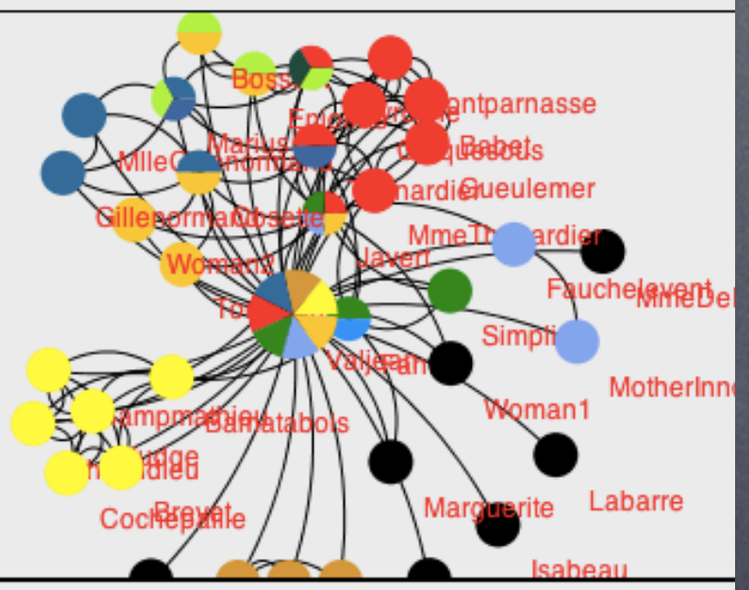
CFinder4



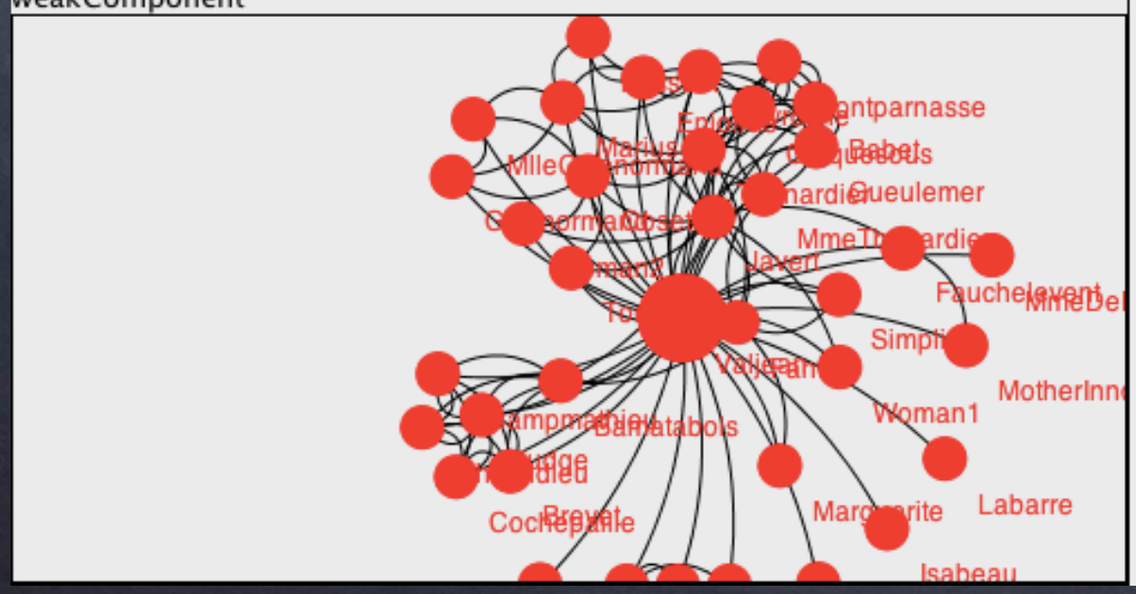
InfoMap



iLCD



weakComponent



Applications

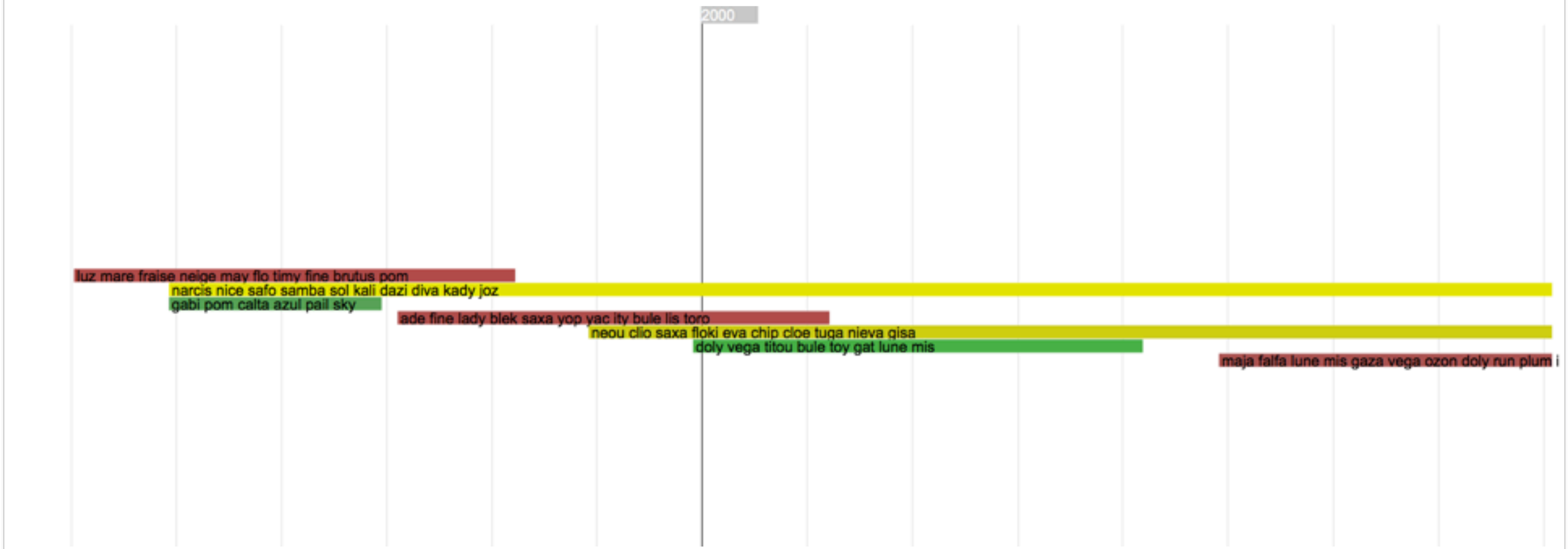
Application 1: Izards

- Working with ethologists
- 20 years of observations (precision: 1 Day)
- Sequences of co-observations

Generation of dynamic network

- For each co-observation
- As long as there is at least N observations on a period P, the link exist.

Results


2000
luz mare fraise neige may flo timy fine brutus pom
narcis nice safo samba sol kali dazi diva kady joz
gabi pom calta azul pail sky
ade fine lady blek saxa yop yac ity bule lis toro
neou clio saxa floki eva chip cloe tuqa nieva gisa
doly vega titou bule toy gat lune mis
maja falfa lune mis gaza vega ozon doly run plum i

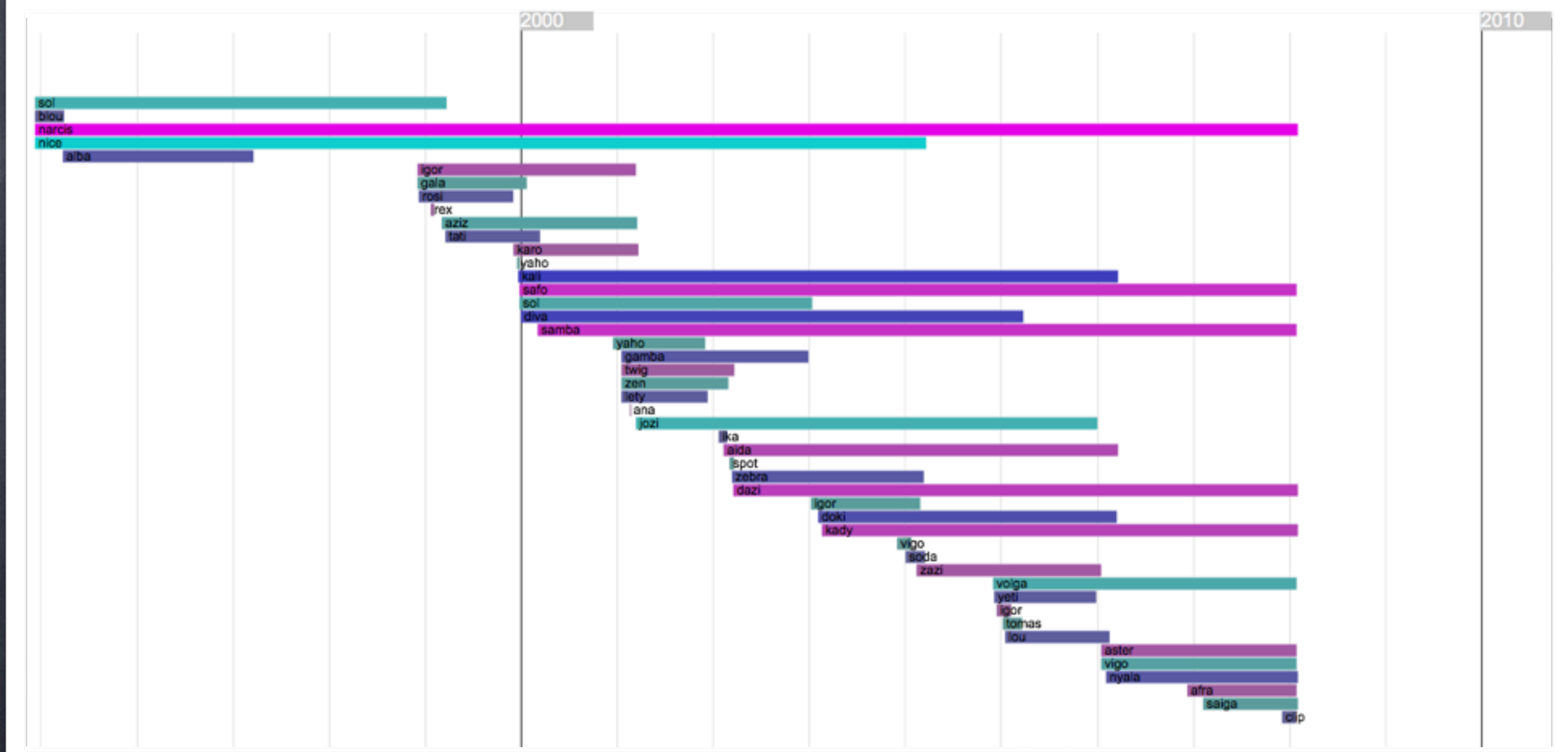
Results

Ethologists question: How strong are communities ?

Previous static analysis : "some individuals switch between groups"

Dynamic communities: "communities are persistent, even when most of original individuals have disappeared"

Results


2000
2010
sol
bleu
narcis
nice
alba
igor
gala
rosi
rex
aziz
tati
karo
yaho
kari
safo
sol
diva
samba
yaho
gamba
twig
zen
ety
ana
jozi
ika
aida
spot
zebra
dazi
igor
doki
kady
vigo
soda
zazi
volga
yeti
igor
tomas
lou
aster
vigo
nyala
afra
saiga
clip

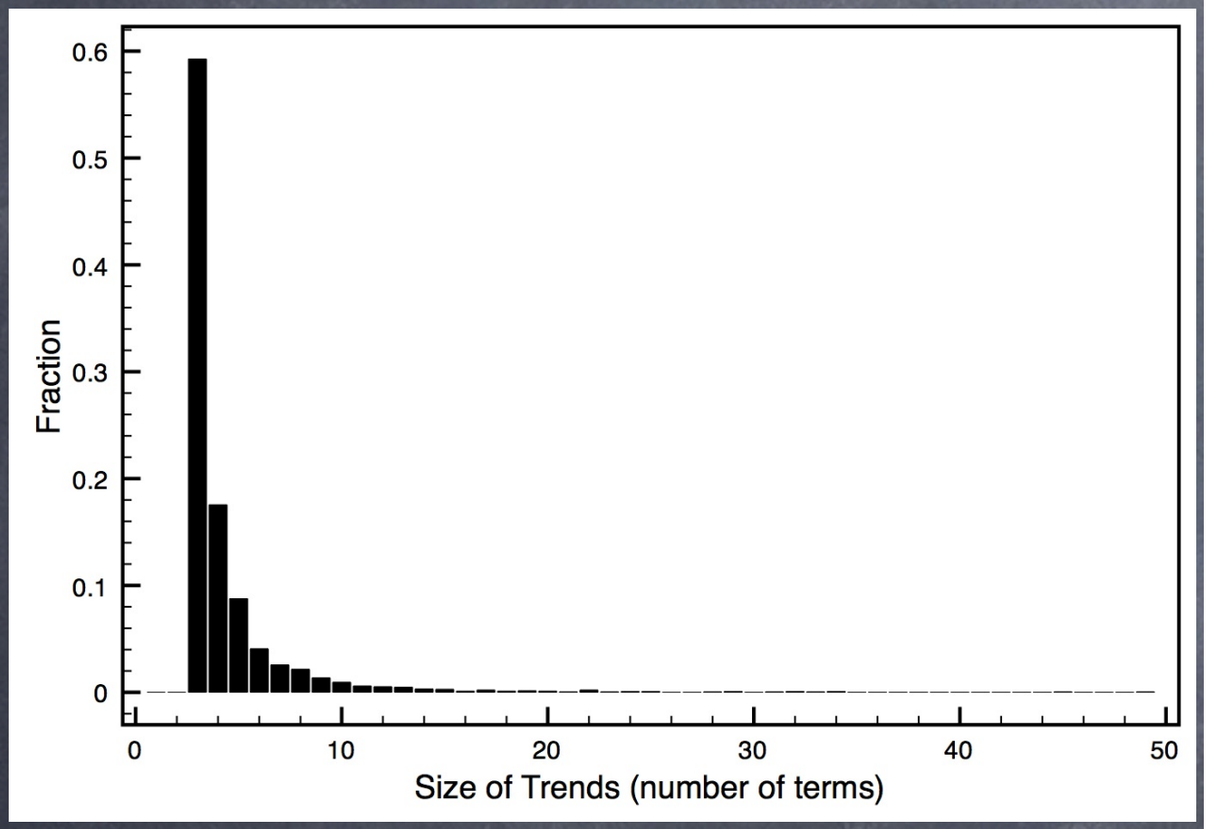
Application 2: Web 2.0 Social Network

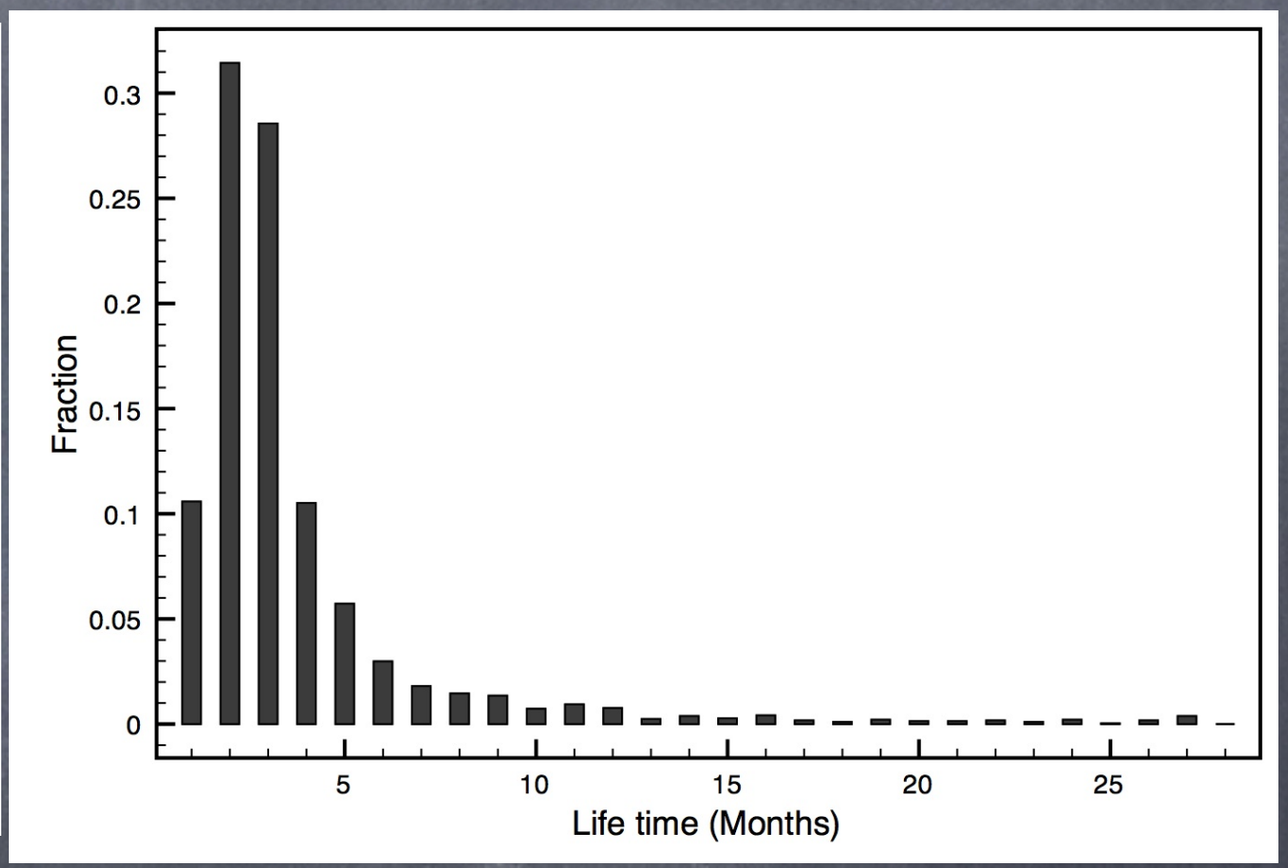
- Nico Nico Douga:
  - Japanese video sharing network
  - 2 years, 3 Million Videos (complete dataset)
  - 1-10 keywords by Video

Application 2: Web 2.0 Social Network

- Evolving network of keywords:
  - If 2 keywords are used simultaneously (co-occurrence) N times on a period P, the link is active
- Community detection = "trend detection"

Results


Fraction
0.6
0.5
0.4
0.3
0.2
0.1
0
0
10
20
30
40
50
Size of Trends (number of terms)


Fraction
0.3
0.25
0.2
0.15
0.1
0.05
0
5
10
15
20
25
Life time (Months)

Results

- Typical communities:
  - Short events

detected event	creation date	ending date	release date
Devil May Cry	12/02/2007	09/08/2008	01/31/2008
Fable 2	12/06/2008	02/03/2009	12/18/2008
GearsOfWar2	10/14/2008	12/29/2008	11/07/2008
Assassin's Creed	01/25/2008	02/26/2008	01/31/2008
Soul Calibur IV	07/07/2008	11/15/2008	07/31/2008
Uncharted	11/11/2007	01/02/2008	11/16/2007

Results

- Typical communities :
  - General topics

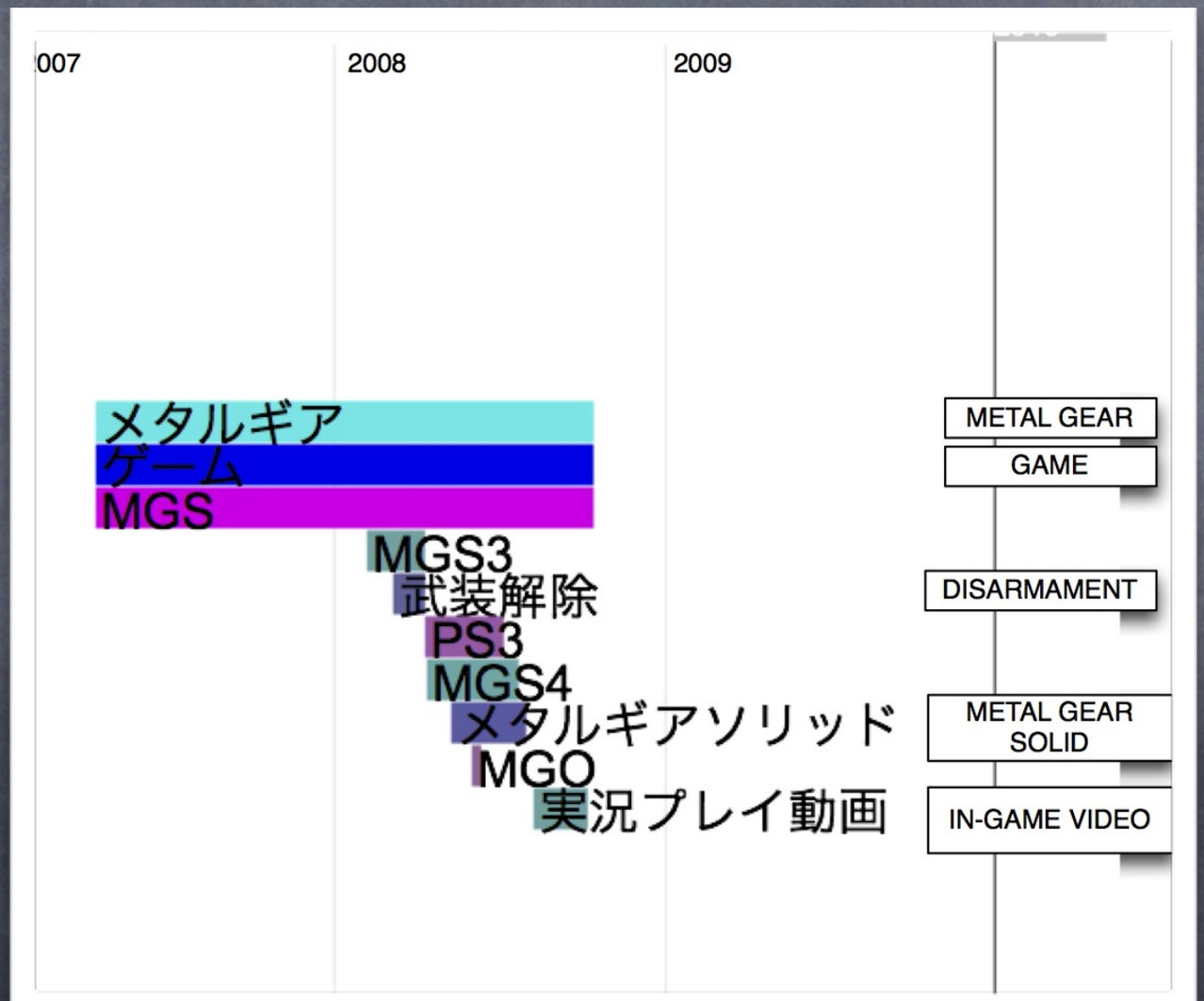
Te i eee	ea i a e	e i g a e
a i a i al i i a i e	3 2	
eag e e	4 13 2	
al a a ge e i i	1 24 2	
e li g F	1 2	1 25 2

Results

- Typical communities :
  - Repetitive Events (Christmas, Tour de France, ...)

Results

- Details of communities' evolution



Conclusion

- Dynamic community detection can be helpful
  - Working on snapshots isn't probably the best choice
- iLCD gives convincing results
  - A lot of possible improvements ! (hierarchy, sparse communities, weights,...)
- All tools developed are (or will be) accessible
  - iLCD (with static/dynamic results)
  - Facebook App (app/source code)
  - Communities explorer
  - Static visualisation of communities
  - Dynamic visualisation of communities