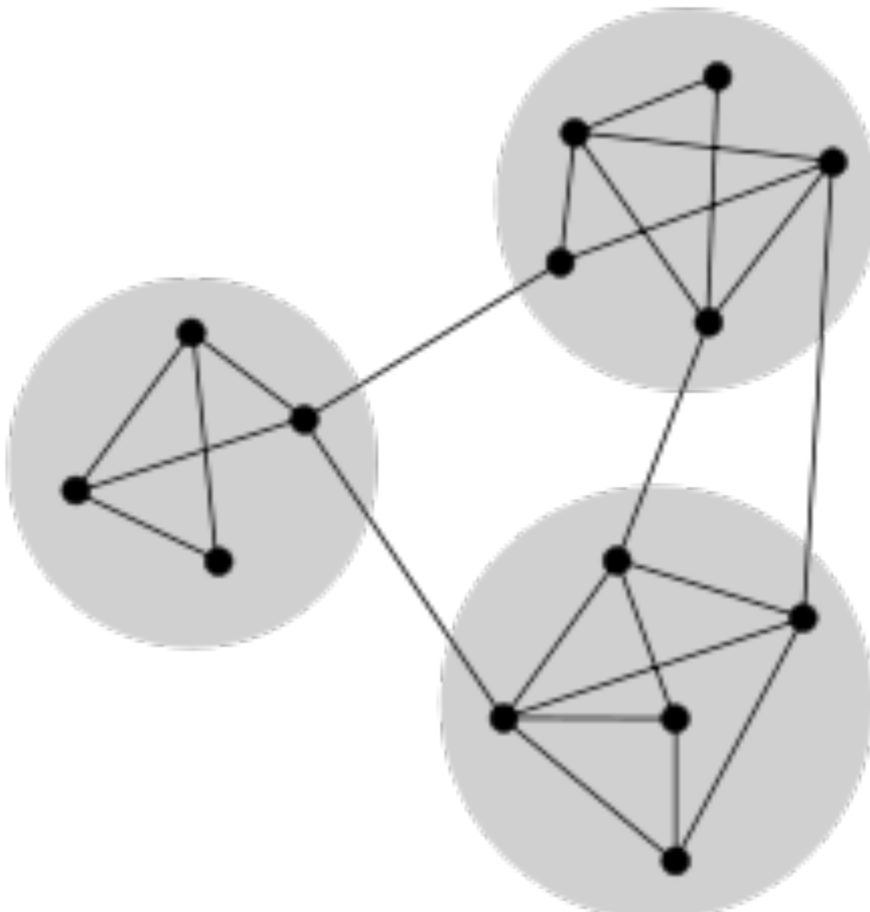
Detection of temporal communities in strongly evolving networks

CAZABET Rémy

Toulouse University
IRIT
France

Université de Toulouse

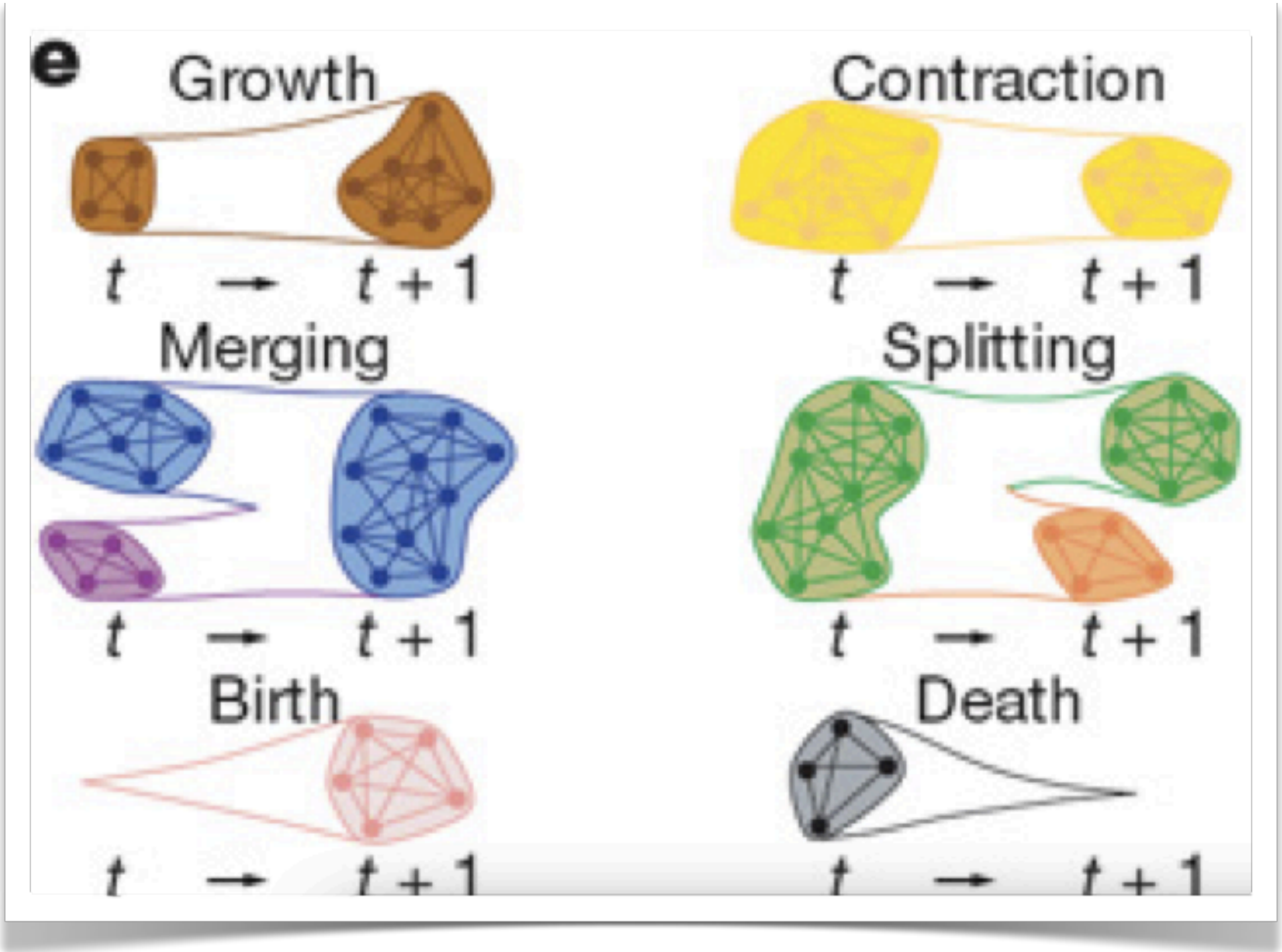
Community detection



Dynamic networks

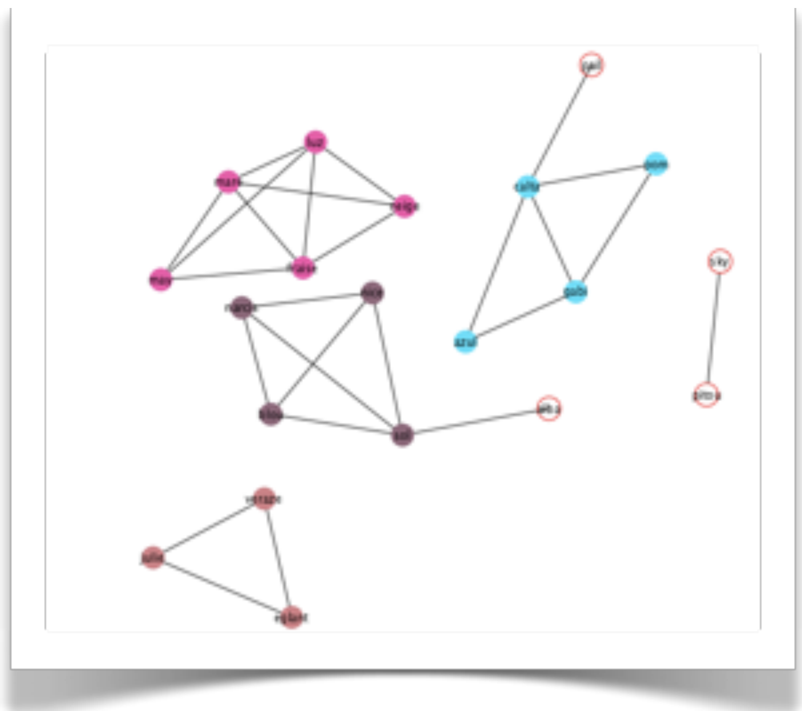
- Traditional view: sequence of snapshots
  - A few snapshots...
- “Strongly evolving networks”
  - All modifications are considered

Temporal communities

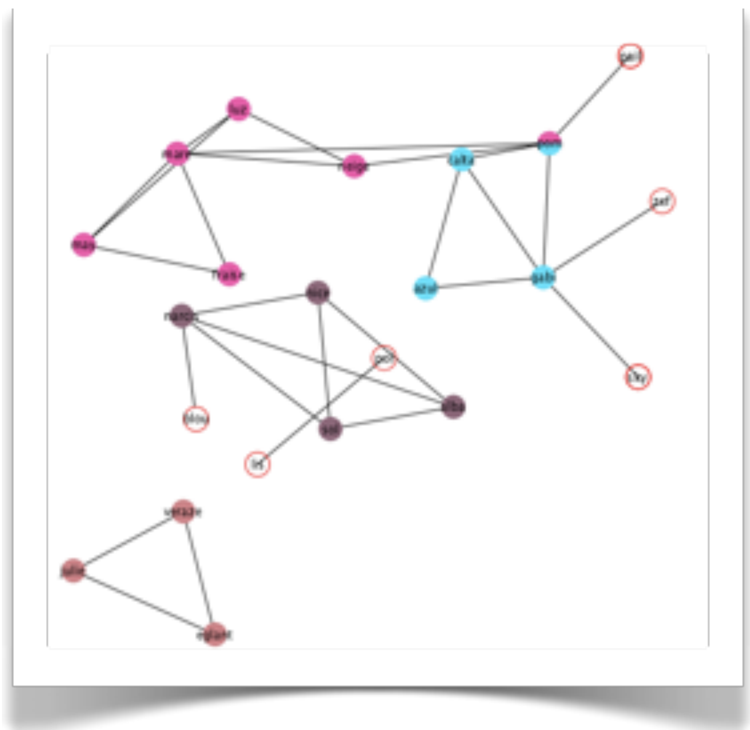


[Palla et al. 2005]

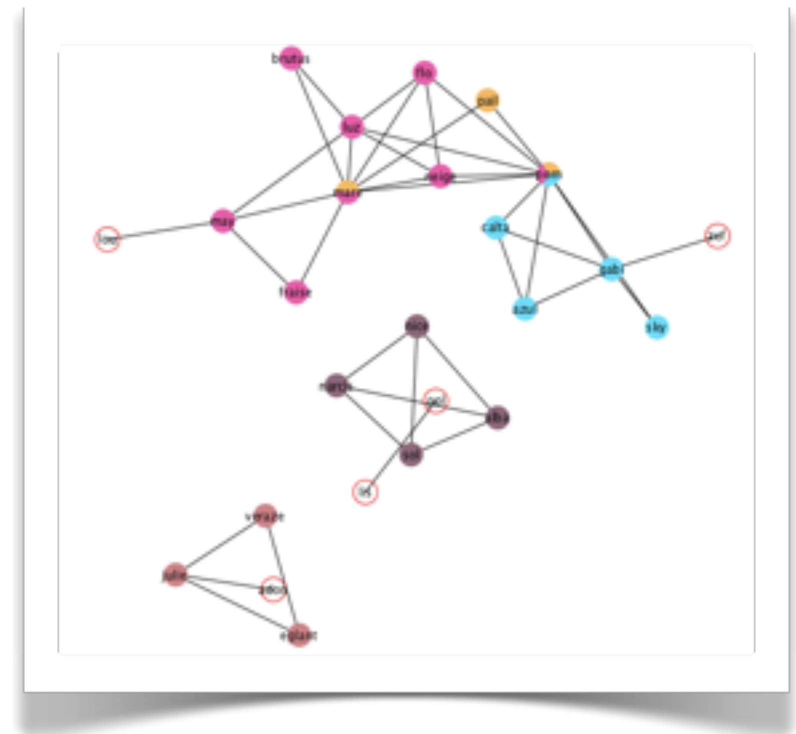
Temporal community detection



T1



T2



T3

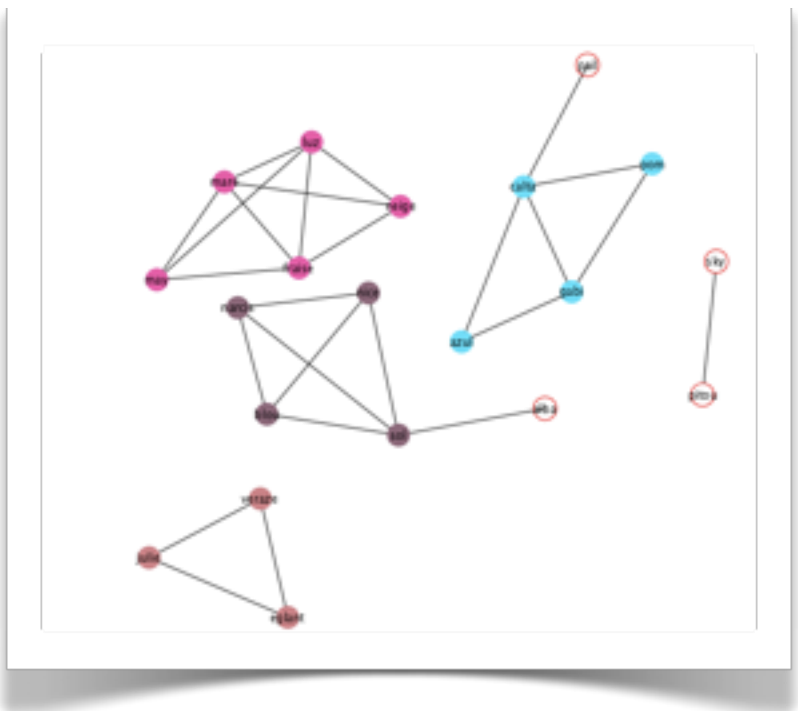
1) Detect communities on snapshots independently
2) Try to match the communities of T+1 with the ones of T

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

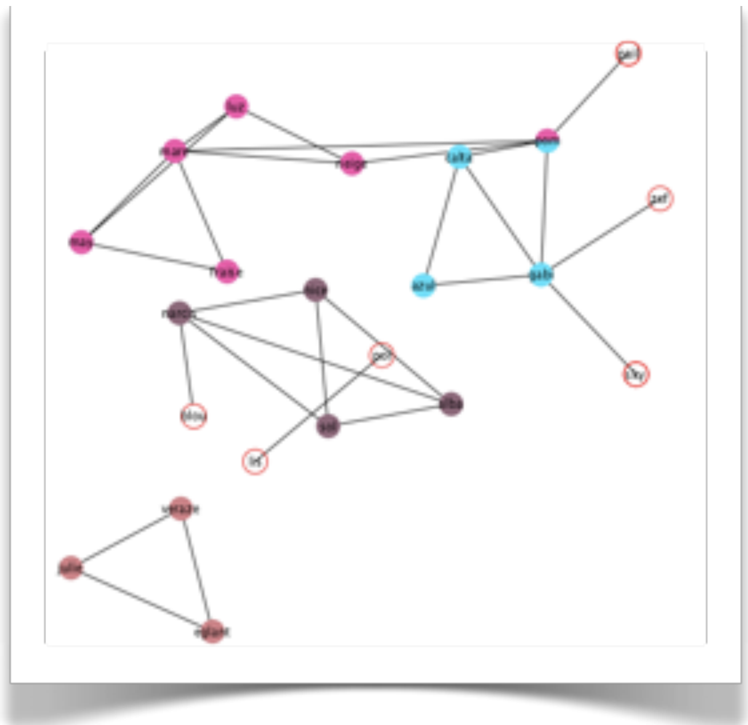
Temporal community detection

- 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

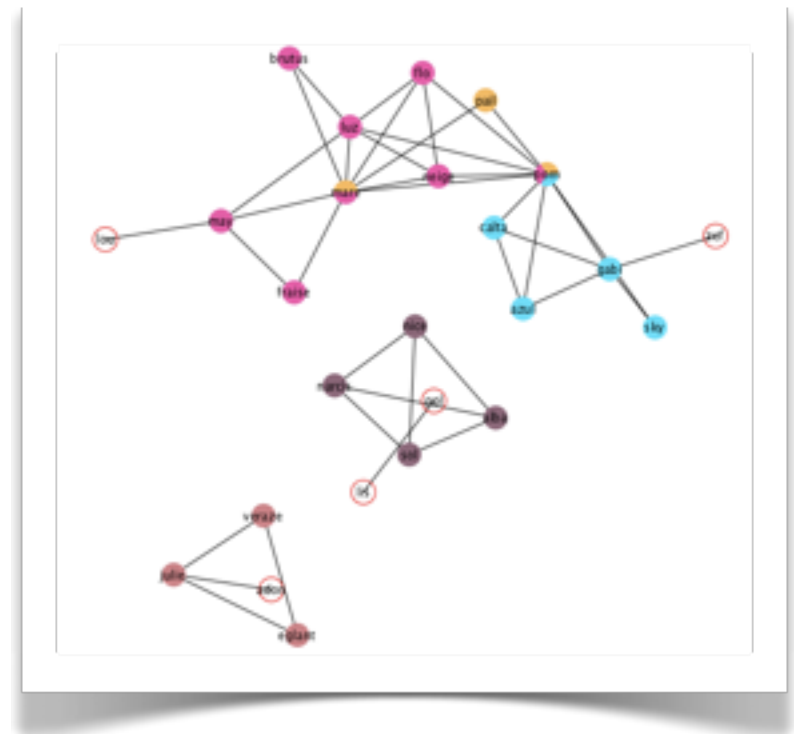
Temporal community detection



T1



T2



T3

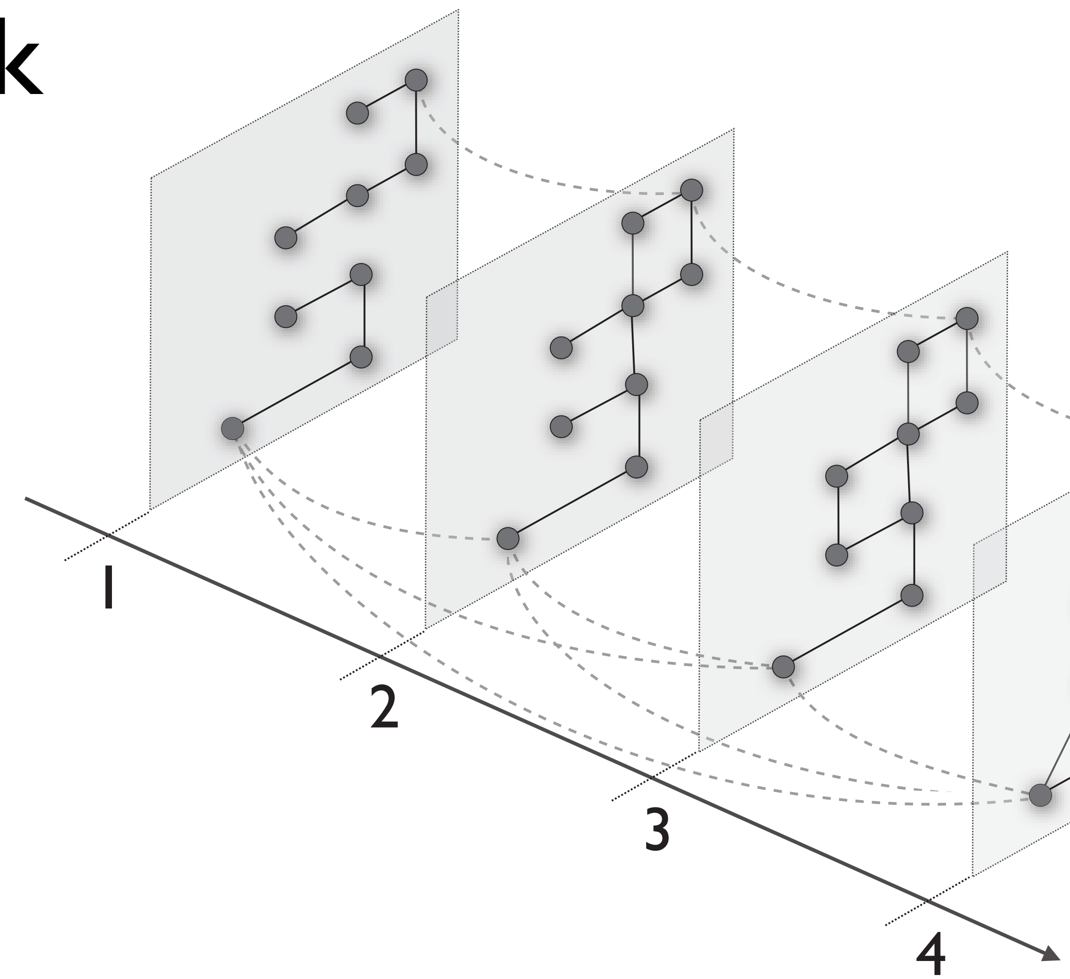
1)Detect communities on the first snapshot
2)Detect communities at T+1 by trying to get good communities while staying coherent with the previous snapshot

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

Temporal community detection

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]

Temporal community detection

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

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)

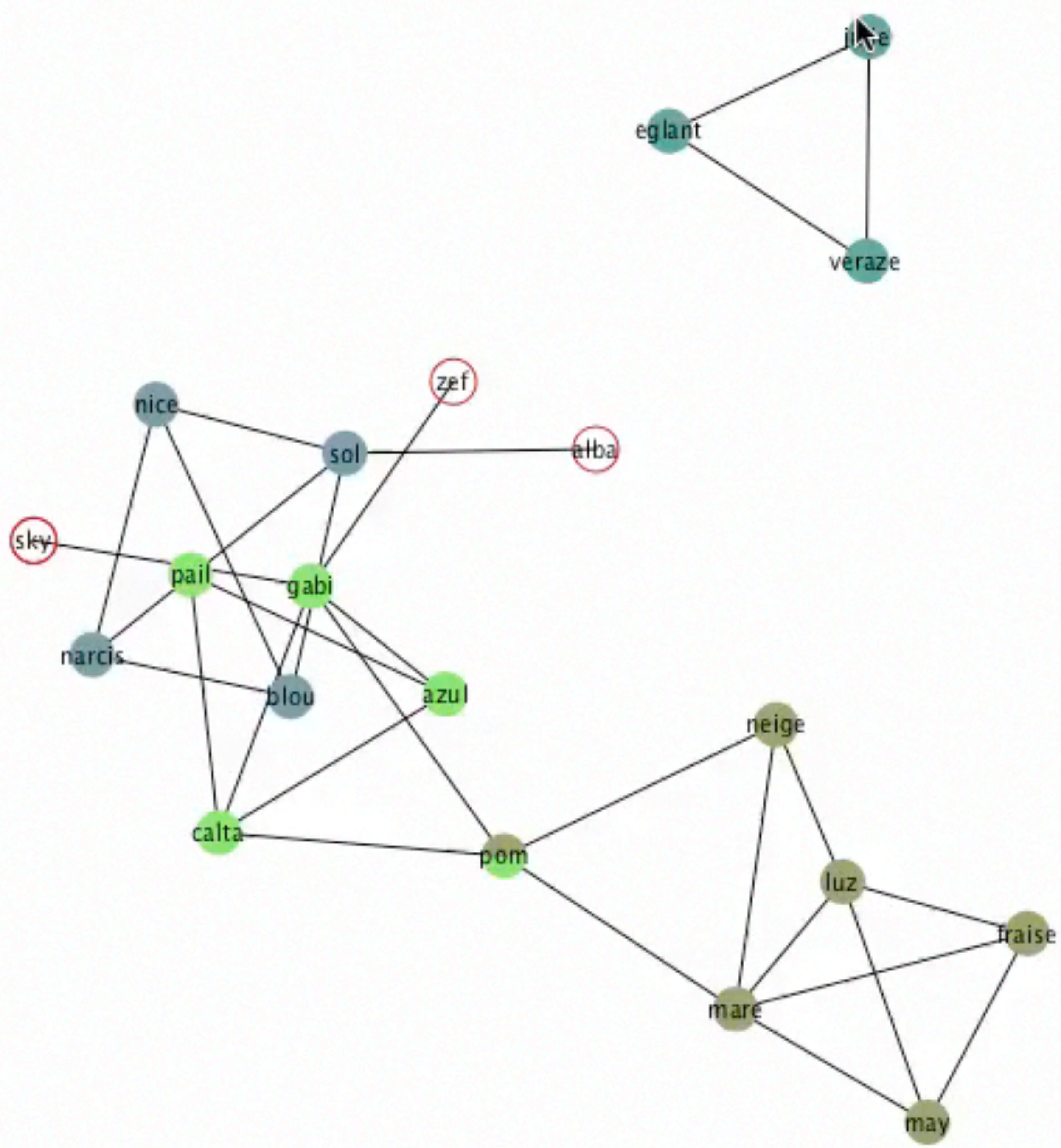
Algorithm

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

Algorithm

- “Play” the evolution of the network
- Communities act as “living entities”: at each step, they can
  - Be born
  - Die
  - Integrate new nodes
  - Reject some nodes

What does it looks 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 clique 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 community c is first computed when the agent node is added to a community. This value is defined as

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

1) Value of seclusion: The value of $seclusion$ of the community is first computed when the community agent is created. This value measures the quality of the community, more precisely how well the community is separated from the remaining part of the system. This value is computed as

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

2) Decide to integrate or not a node agent: When a community agent c receives a request from a node agent n asking for his integration, it first computes its potential belonging, $pb(c, n)$. This value is defined as

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

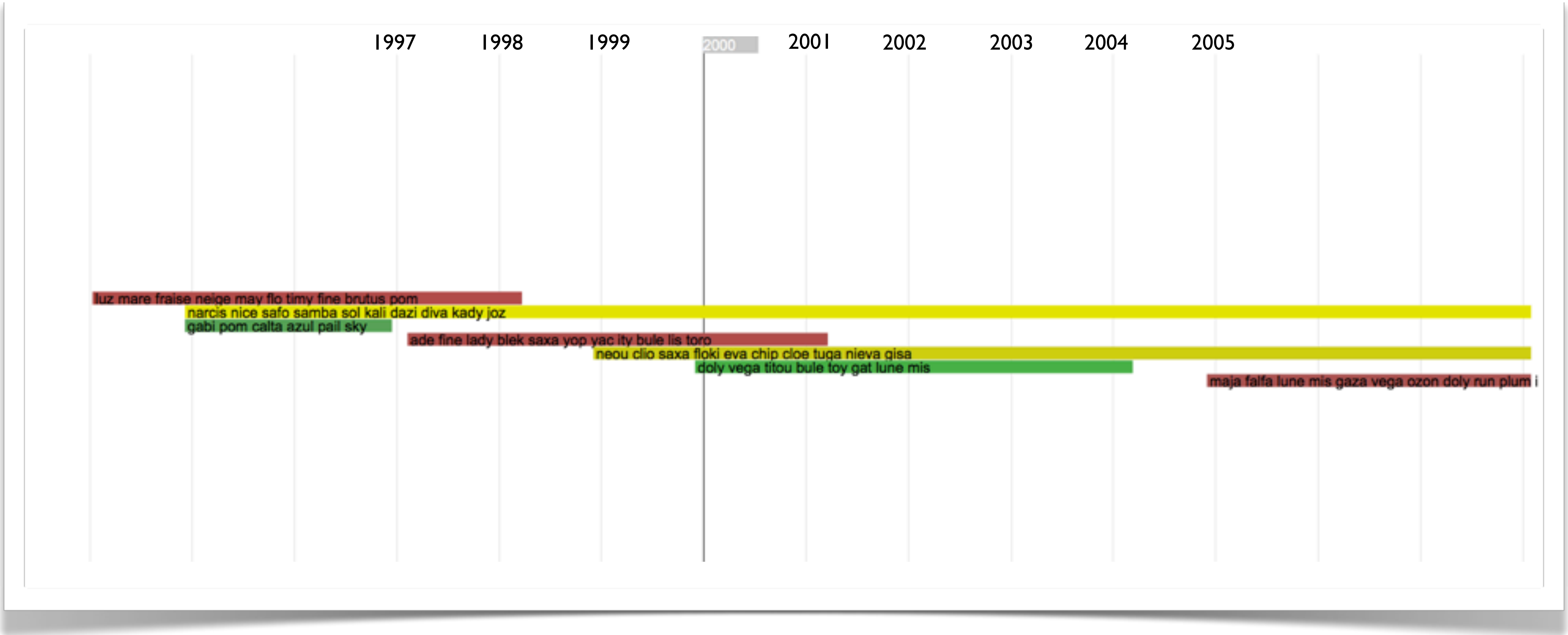
Application: Izards

- 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


1997
1998
1999
2000
2001
2002
2003
2004
2005
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 tuga nieva gisa
doly vega titou bule toy gat lune mis
maja falfa lune mis gaza vega ozon doly run plum

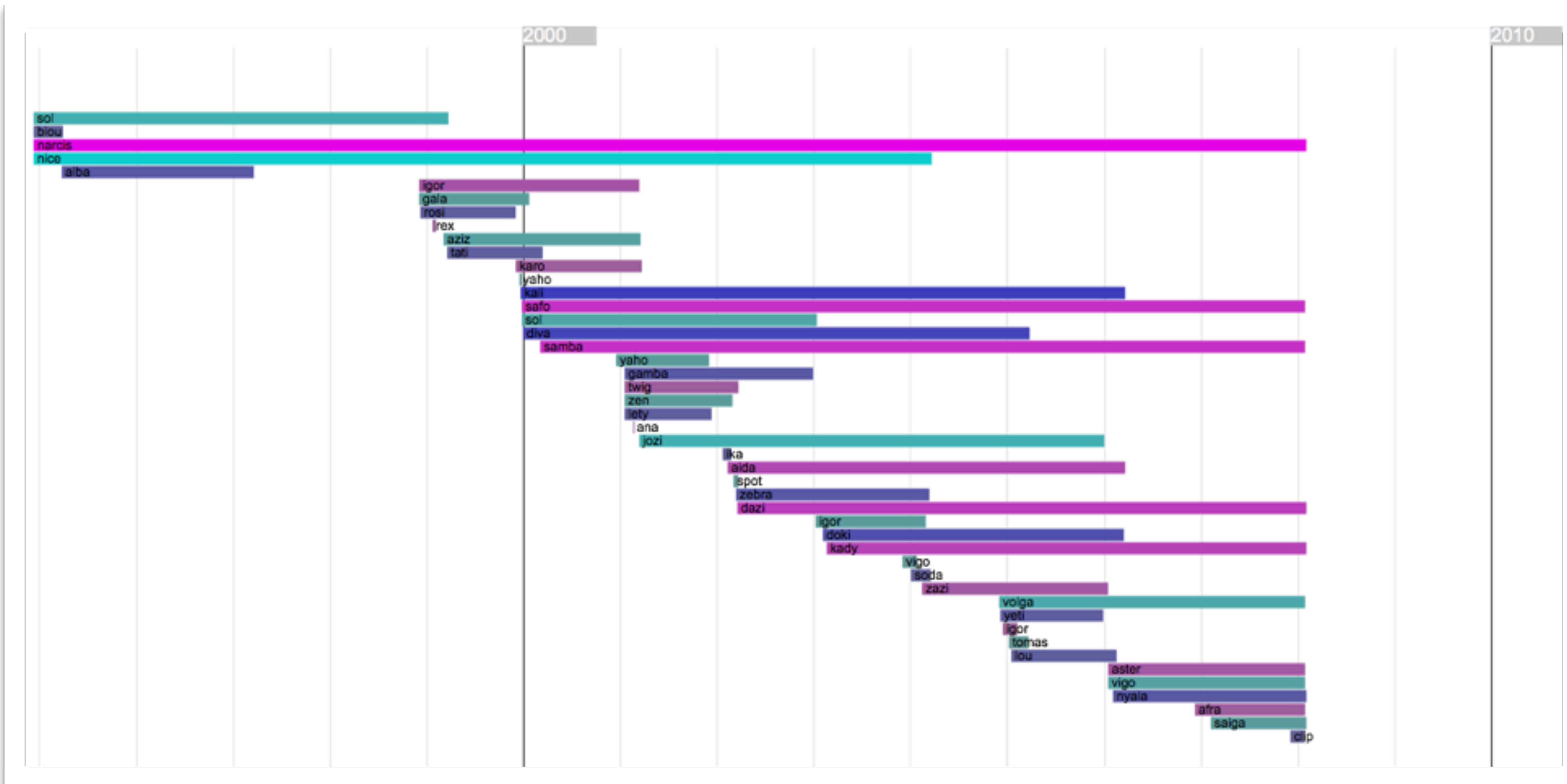
Results

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
biou
narcis
nico
alba
igor
gala
rosi
rex
aziz
tati
karo
yaho
kali
safo
sol
diva
samba
yaho
gamba
twig
zen
loty
ana
jozi
ika
aida
spot
zebra
dazi
igor
doki
kady
vigo
soda
zazi
volga
yeti
igor
tomas
lou
aster
vigo
nyala
afra
saiga
clip

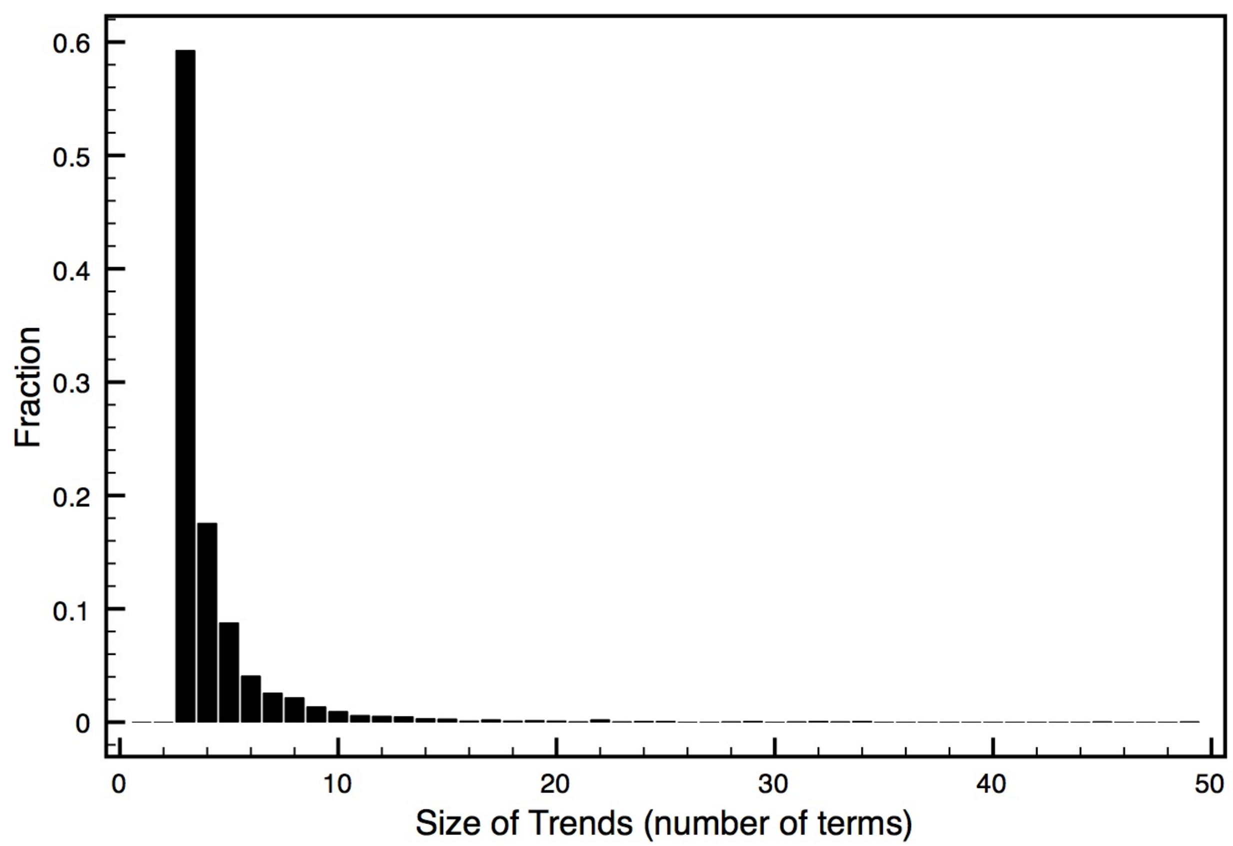
Application: Web 2.0 Social Network

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

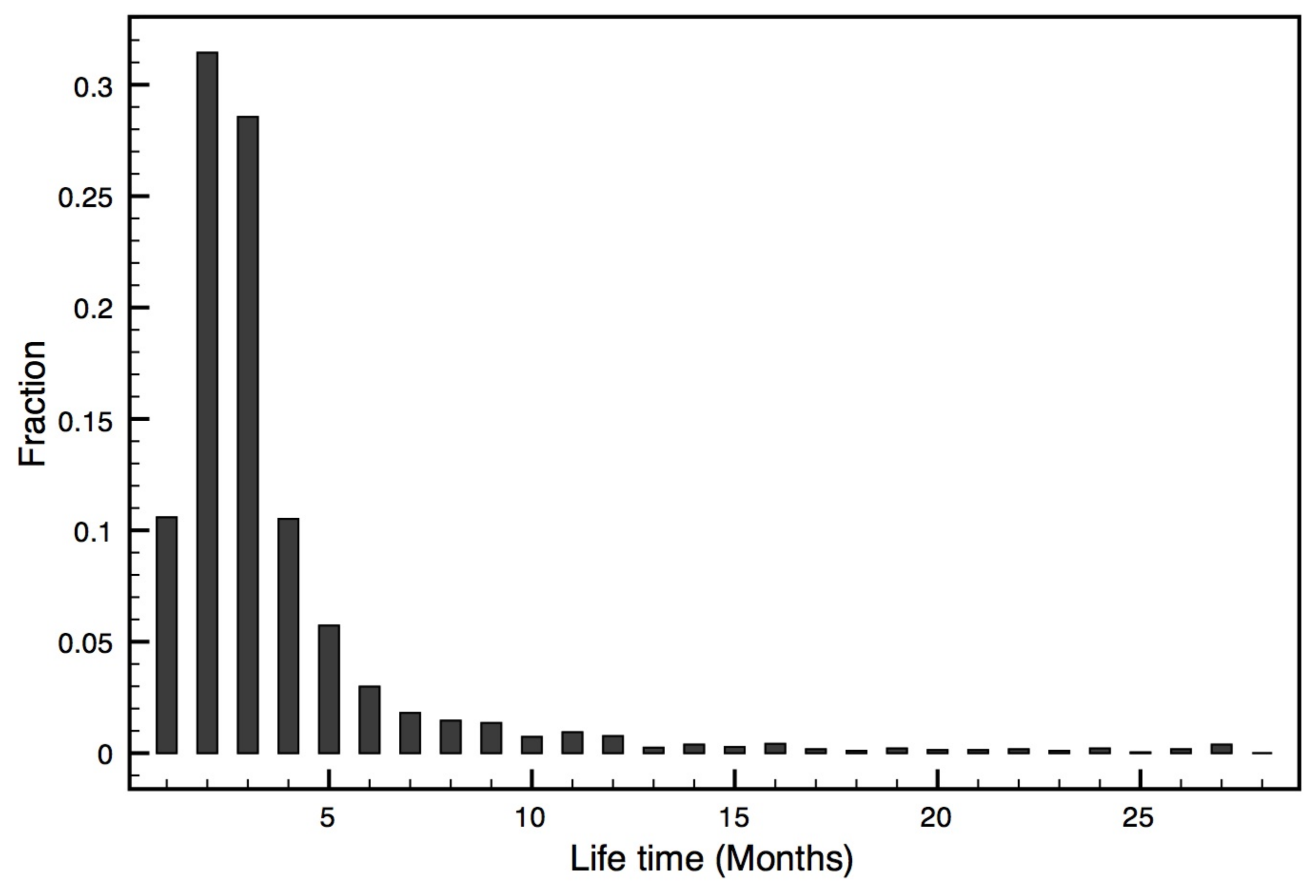
Application: Web 2.0 Social Network

- 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



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



0.3
0.25
0.2
0.15
0.1
0.05
0
Fraction
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

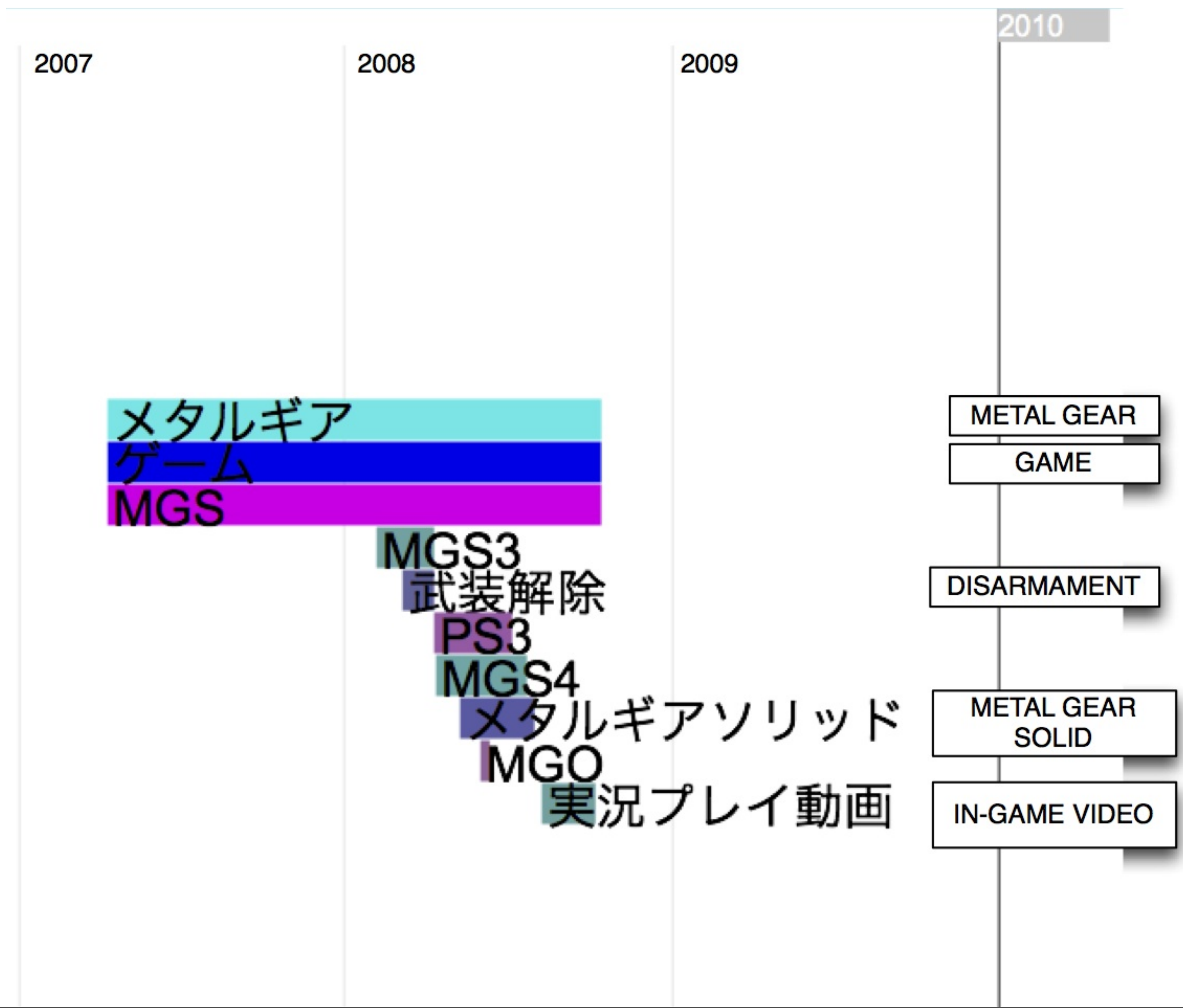
Terms in the event	creation date	ending date
Cats, kitty, animal, Nico Nico Cats' videos	03/06/2007	-
J-League, Soccer, Sport	04/13/2007	-
Vocal, arrangement, Dojin Music	01/24/2008	-
Pro Wresling, WWE,WWF, Sport	07/18/2007	10/25/2008

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
- A lot of possible improvements ! (hierarchy, sparse communities, weights,...)