Community structure : evaluation and motif analysis in link streams

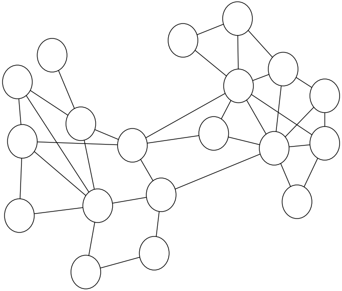
Jean Creusefond, GREYC, Normandy University
Work with : Sylvain Peyronnet, Thomas Largillier, Remy Cazabet
LIP6 Presentation

May 12, 2016

Ground-truths and quality functions

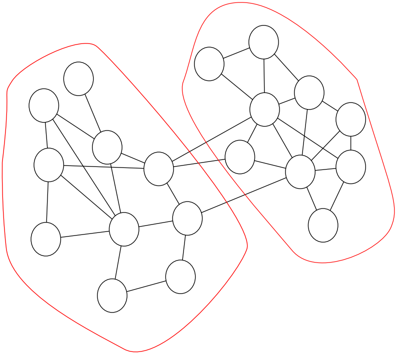
Community detection and quality functions

Social network :



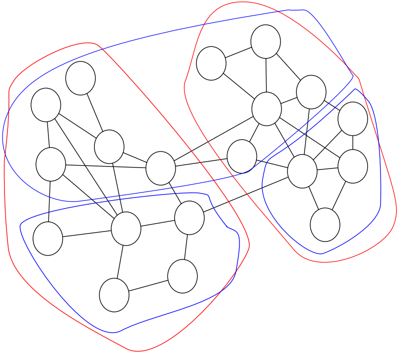
Community detection and quality functions

A clustering :



Community detection and quality functions

Two clusterings :



How to chose the best?

We use **quality functions**, for optimisation and evaluation

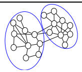
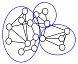
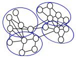
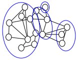
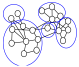
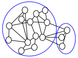
Algorithms application

Alg1
Alg2
Alg3

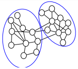
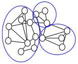
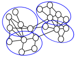
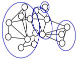
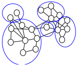
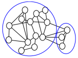
×

Graph1
Graph2

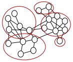
⇒

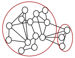
	Graph1	Graph2
Alg1		
Alg2		
Alg3		

Comparison with ground-truth

	Graph1	Graph2
Alg1		
Alg2		
Alg3		

+

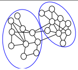
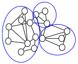
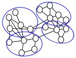
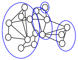
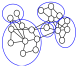
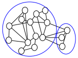
Ground-truth1 : 

Ground-truth2 : 

⇒

	Graph1	Graph2
Alg1	gold-standard :	...
Alg2	trustable	...
Alg3	value	...

Application of quality functions

	Graph1	Graph2
Alg1		
Alg2		
Alg3		

$\times$ Quality1, Quality2, Quality3 $\Rightarrow$

	Quality1		Quality2		Quality3	
	Graph1	Graph2	Graph1	Graph2	Graph1	Graph2
Alg1	quality score	...				
Alg2	...	...				
Alg3	...					

Coherence quantification

	Quality1		Quality2		Quality3	
	Graph1	Graph2	Graph1	Graph2	Graph1	Graph2
Alg1	quality score	...				
Alg2	...	...				
Alg3						

/

	Graph1	Graph2
Alg1	gold-standard :	...
Alg2	trustable	...
Alg3	value	...

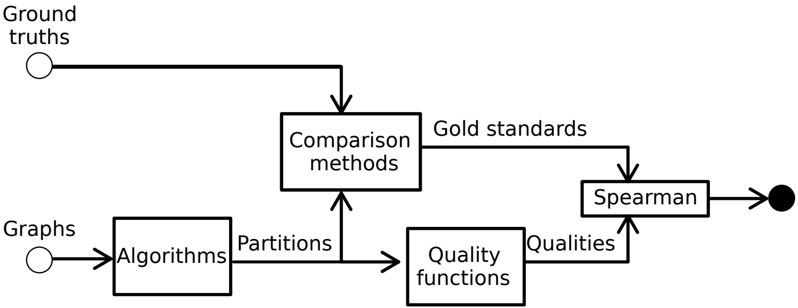
⇒

	Quality1	Quality2	Quality3
Graph1			
Graph2			

red : good coherence

blue : bad coherence

Methodology



We measure the coherence of the quality functions with ground-truth data

Finding context

Context : ground-truths where quality functions behave the same way.

	Quality1	Quality2	Quality3
Graph1			
Graph2			

⇒

	Graph1	Graph2
Graph1	-	
Graph2	-	-

Need for data

Ground-truths	Communities that can be trusted
Algorithms	They should have various designs.
Quality functions	The main items to compare
Comparison methods	Multiple functions output complementary results

Comparison methods

Normalised Mutual Information (NMI)

Captures the quantity of information needed to infer one clustering from the other.

F-BCubed

The average ratio, over all individuals, of neighbors in one clustering that are still neighbors in the other one.

Synthetic networks

Is the methodology able to recognise networks with a similar structure?
LFR benchmark : tunable virtual graphs, with social-network structure.

Results

- NMI : Globally coherent with our expectations, but influenced by random generation
- F-BCubed : More robust, difference of overlapping over-matches

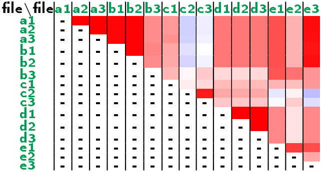


Table 1 : NMI

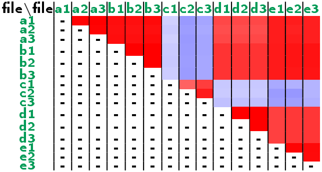
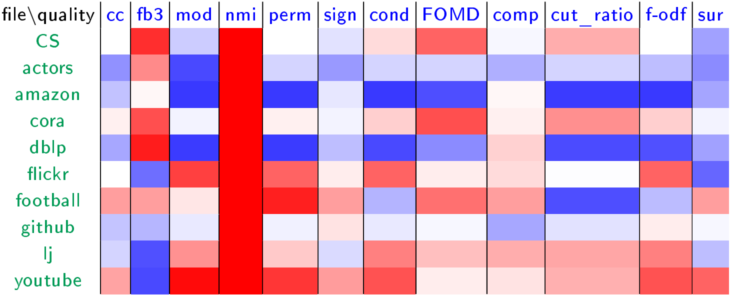


Table 2 : F-BCubed

Results

(flickr, lj, youtube) : Online social networks



(flickr, lj, youtube) : Modularity

CoDACom : free software

CoDACom

Home | Install | How to use | How to extend | Contact | Demo | Licence

- A community detection tool
- Efficient community detection
- Ground-truth analysis
- Plug and play
- Extension
- Visualisation
- Modularity

A community detection tool

CoDACom (Community Detection Algorithm Comparator) is a software which purpose is to simplify the research in community detection. It is designed to :

- Run multiple community detection algorithms on graphs
- Output a set of statistics on these graphs and on the runs, from degree distribution to the quality of the different communities
- Take into account ground-truth by comparing it with the results of community detection
- Include home-made implementations of (potentially new) community detection algorithms with no code re-writing
- Include user-written quality functions
- Include user-written extrinsic comparison functions

Efficient community detection

The community detection implementations featured in CoDACom are, for the most part, written in C/C++. For the fastest (loglinear), it is entirely possible to analyse graphs with millions of edges on a standard desktop in less than an hour. Using a server grants the advantage of running all the couples (methods,

Communities and temporal motifs

Definition

Hypothesis

The communication structure (i.e. frequent motifs) is different inside and outside communities.

A **motif** is a regularly repeated communication pattern. A motif has a depth (distance from origin), a size (number of nodes) and a level (number of edges).

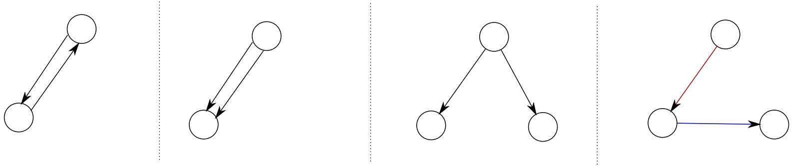
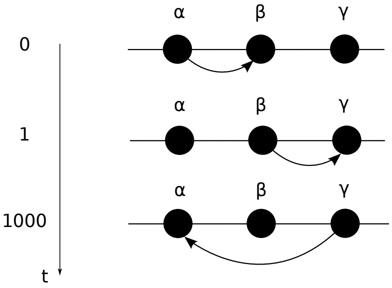


Figure 1 : All level 2 motifs

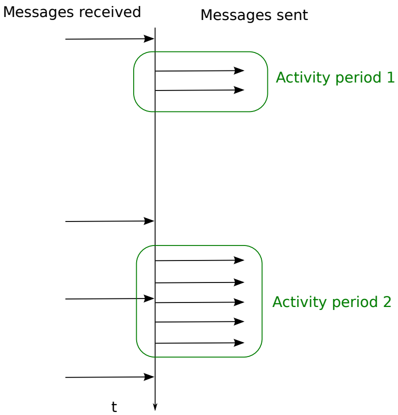
Traditionnally (Zhao et al. 2010, Tabourier et al. 2012), causality is assessed if there is a small time (parameter W) difference between emission of the messages.



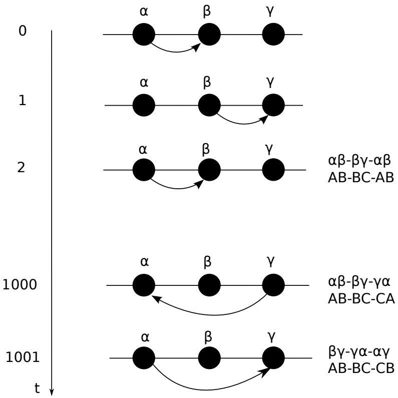
Not adapted to asynchronous communication.

In this example, AB-AC-CA is not a motif if $W < 999$.
What if C was just away?

Activity periods



Example of activity period-based motifs



Null models

A null model, generated from a base graph, is the same except for a structural property that has been randomised.

Objective

By differentiating the null model and the base graph, one can isolate the influence of the randomised property.

Time-mixing model : the timestamps of communications are shuffled for each user $\Rightarrow$ causality is destroyed

Using the time-mixing model to asses the influence of causality

Assumption : the measured values follow a gaussian distribution in the null model (checked in practice).
Therefore :

- low ($\sim 0.3\%$) probability that a value from this distribution would be far from the average ($3 \times \sigma$)
- a point far from the average probably does not come from this distribution
- the distribution difference is due to the destruction of causality (the only property that is randomised)

Experiments : data

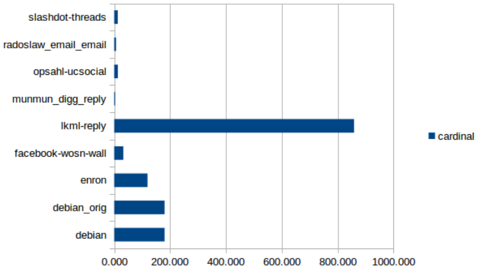
Data from KONECT (http://konect.uni-koblenz.de/):

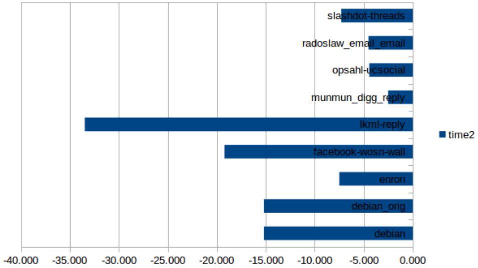
- Digg : reply-to
- lkml : reply-to
- slashdot : reply-to
- radoslaw : mail
- Enron : mail
- Facebook wall : post-to
- UC Irvine : instant message

Memberships : iLCD, Louvain and infomap (on aggregated networks)
We also have access to a reply-to network with an overlapping group structure (threads):

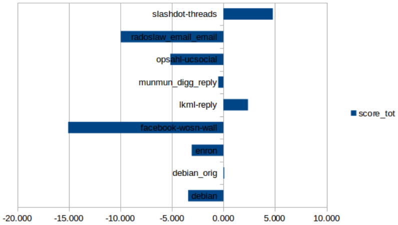
- Debian : reply-to (with membership)

Experiments : temporal triangles (1)





Experiments : temporal triangles (2)



Triangles are more frequent and short than the null model $\Rightarrow$ we are detecting structure
Slightly less included in communities, multiple possible explanation :

- individuals inside of communities use various means of communication
- peripherical communications need structure

Conclusions

No structure inside communities?

No pattern has a surprisingly high community score, except with debian membership.
Negociation with my university to get anonymised data about mails and users.

Using motifs for user categorisation

Analysis at user level : if a user emits a lot of some patterns, does that imply a role for him?