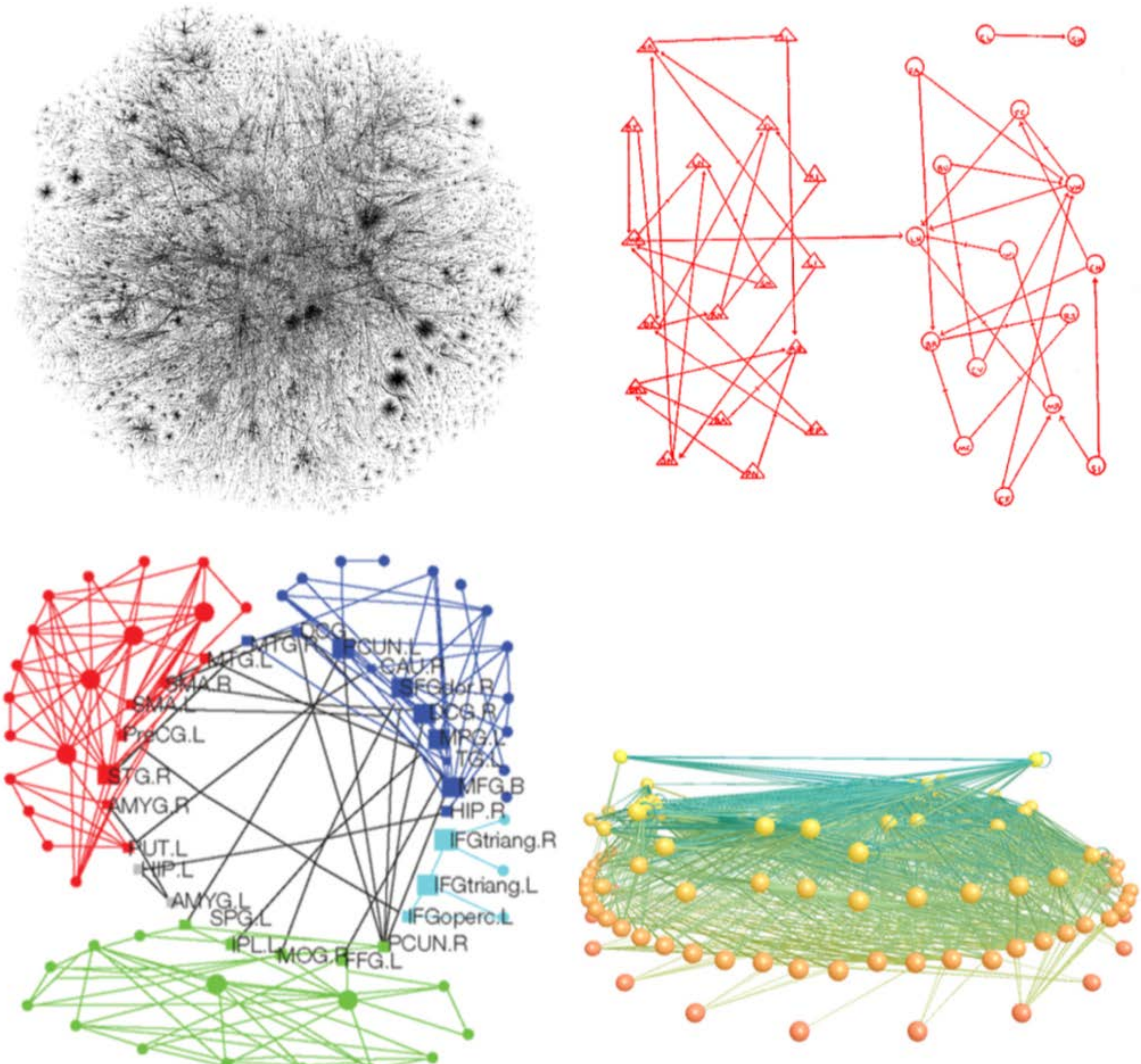
Modularity and Dynamics on Networks

Mathematics of Large Networks

R. Lambiotte
renaud.lambiotte@maths.ox.ac.uk

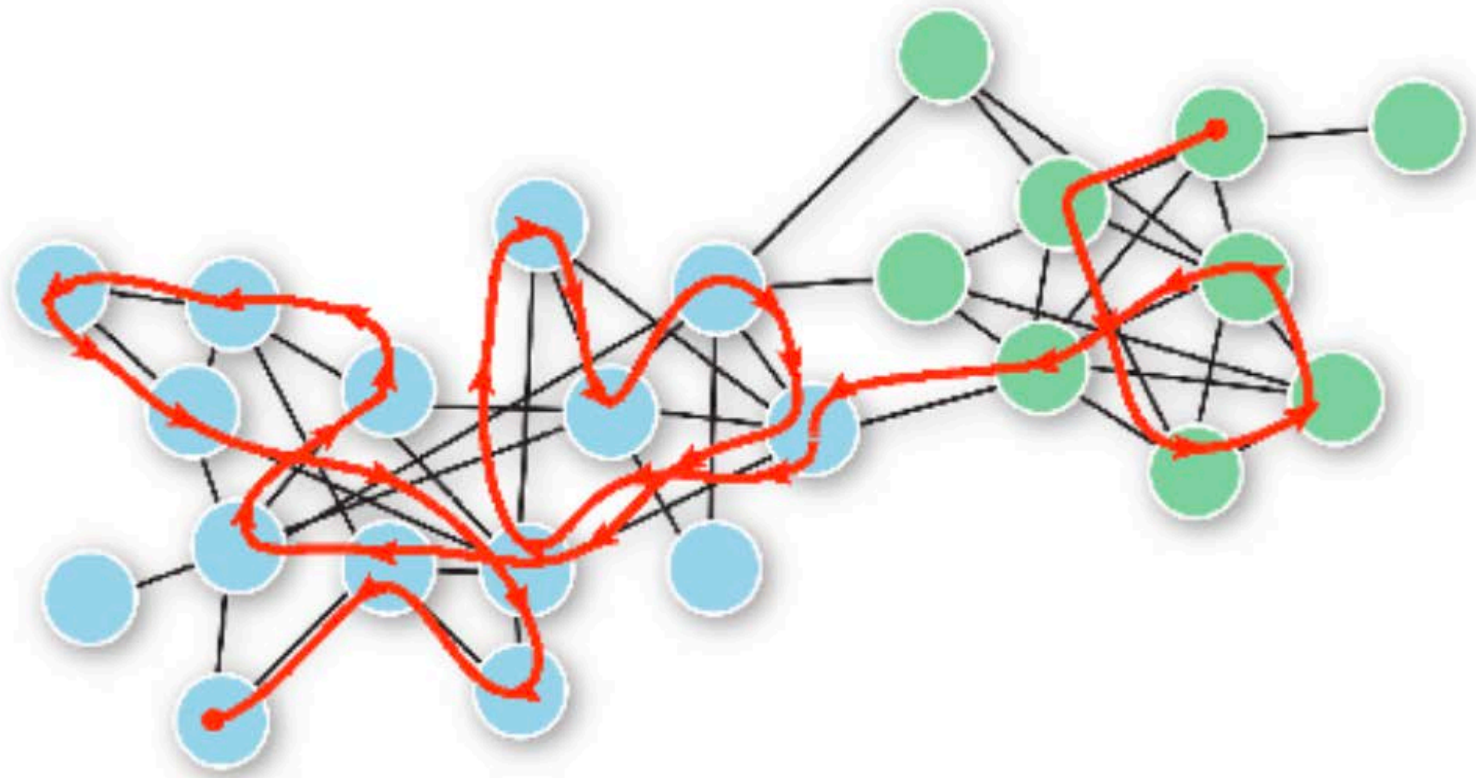

MTG.R
PCUN.L
MTG.L
CAU.R
SMA.R
SFGdor.R
SMA.L
DCG.R
PreCG.L
TG.L
STG.R
MFG.B
AMYG.R
HIP.R
IFGtriang.R
PUT.L
HIP.L
IFGtriang.L
AMYG.L
SPG.L
IFGoperc.L
IPL.L
PCUN.R
MOG.R
FFG.L

Relation between structure and (linear) dynamics

$$\mu \frac{\mathrm{d}\mathbf{x}}{\mathrm{d}t} = L\mathbf{x}$$

Effect of structure on spreading:
SI, random walk, consensus, etc.

Uncover structure from dynamics:
Pagerank, Markov stability, etc.



(1) How does the modular structure of a network affect dynamics?
(2) How can dynamics help us characterise and uncover the modular structure of a network?

Organisation

H1: Community detection and modularity

H2: Time scale separation and modularity

H3: Diffusive dynamics to uncover communities in networks

Problem sheets

Combination of numerics and analytics

Mathematics of Large Networks
Problem Sheet

Renaud Lambiotte and Erik Hormann

FIRST HOUR

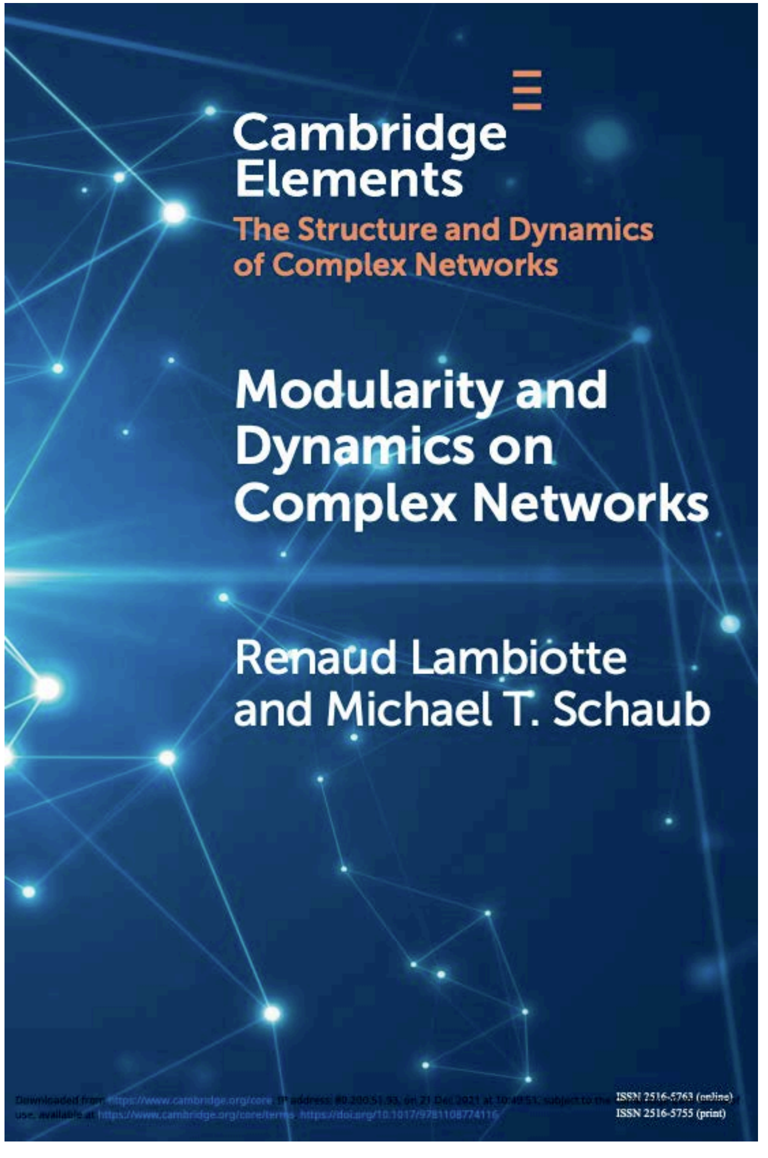
1. *Graph Laplacians.*

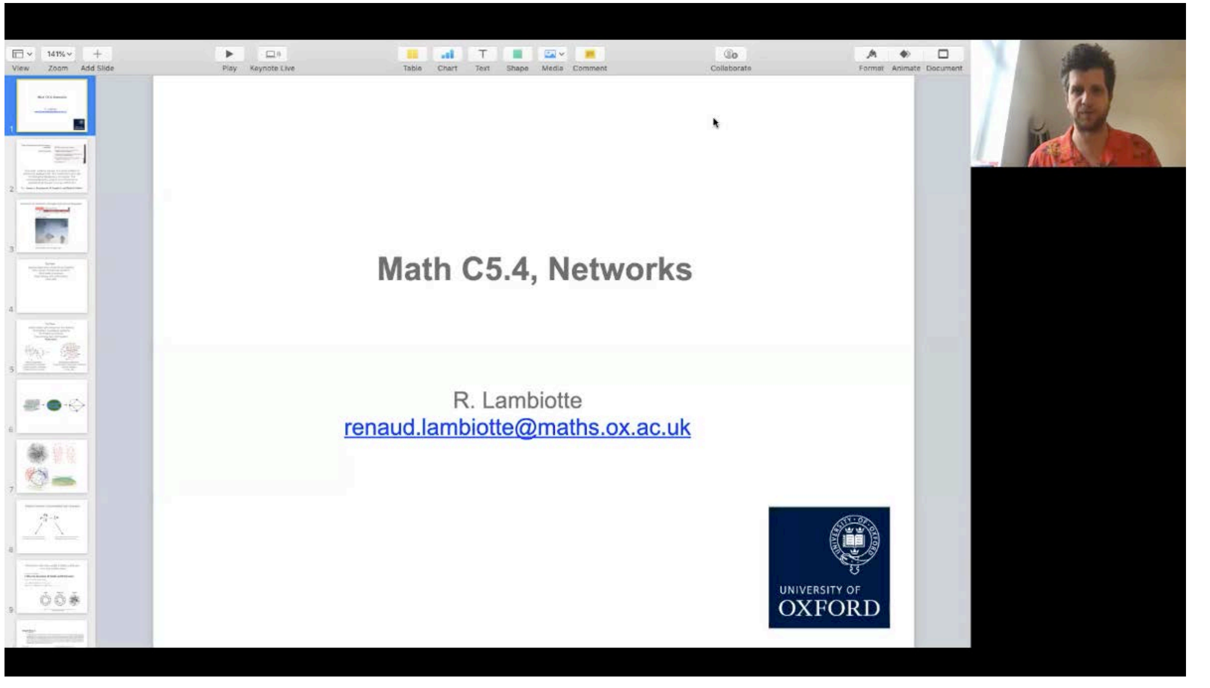
   Consider an unweighted, undirected, simple network. Show that the smallest eigenvalue of the combinatorial graph Laplacian $\mathbf{L} = \mathbf{D} - \mathbf{A}$ is 0. How can one use the spectrum of the graph Laplacian to determine the number of components in the network? Do you have any ideas about how one might think about a graph that is "almost" separated into two disjoint components (and how one might measure how close the components are to being disconnected)?

2. *Modularity*

   (a) Apply modularity optimization techniques implemented in the library of your choice on some examples and visualise the results.

Additional resources





https://www.dropbox.com/s/qtgr4s552e996pg/modularity-and-dynamics-on-complex-networks.pdf?dl=
https://www.youtube.com/watch?v=TQKgB0RnjeY&list=PL4d5ZtfQonW0MsGE4Pn12rxUprPXB4_VS

Organisation

H1: Community detection and modularity

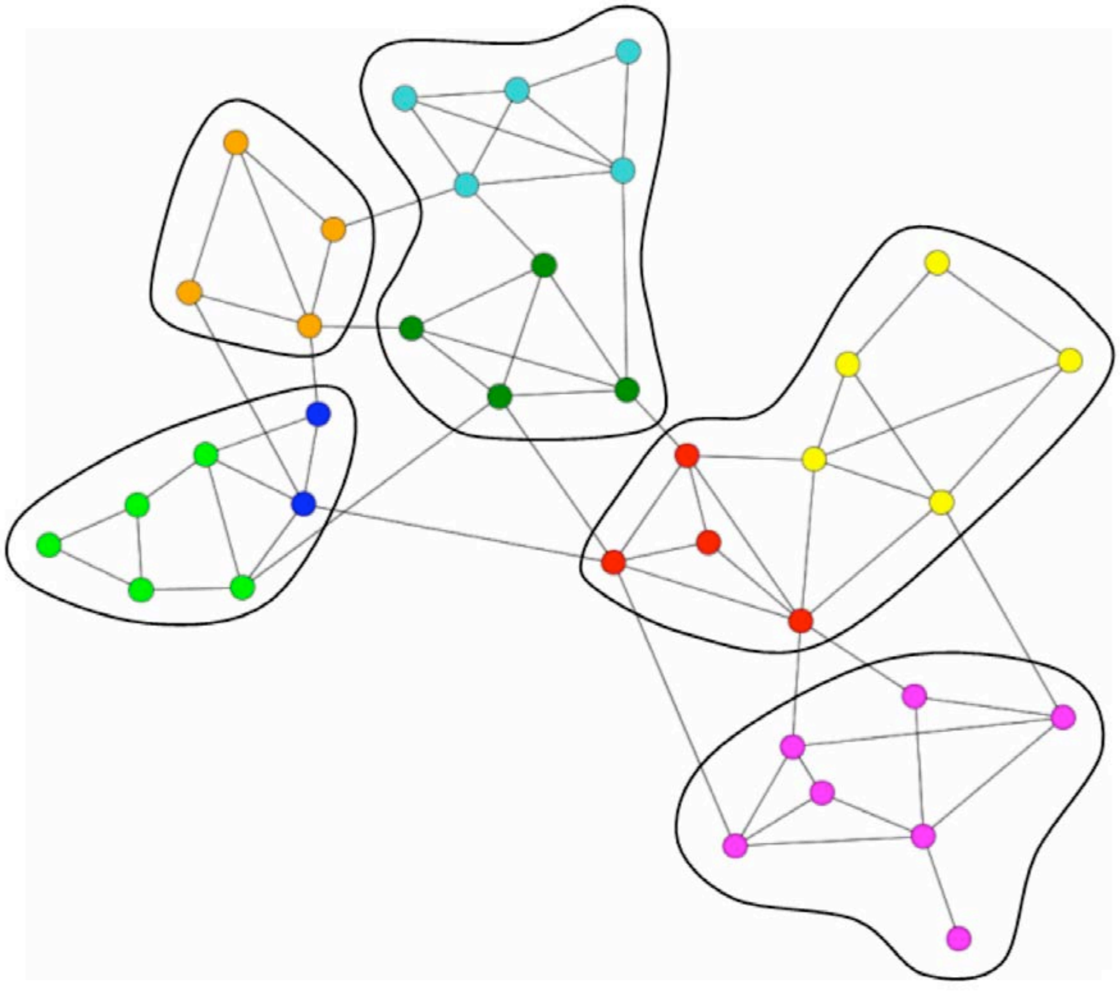
H2: Time scale separation and modularity

H3: Diffusive dynamics to uncover communities in networks

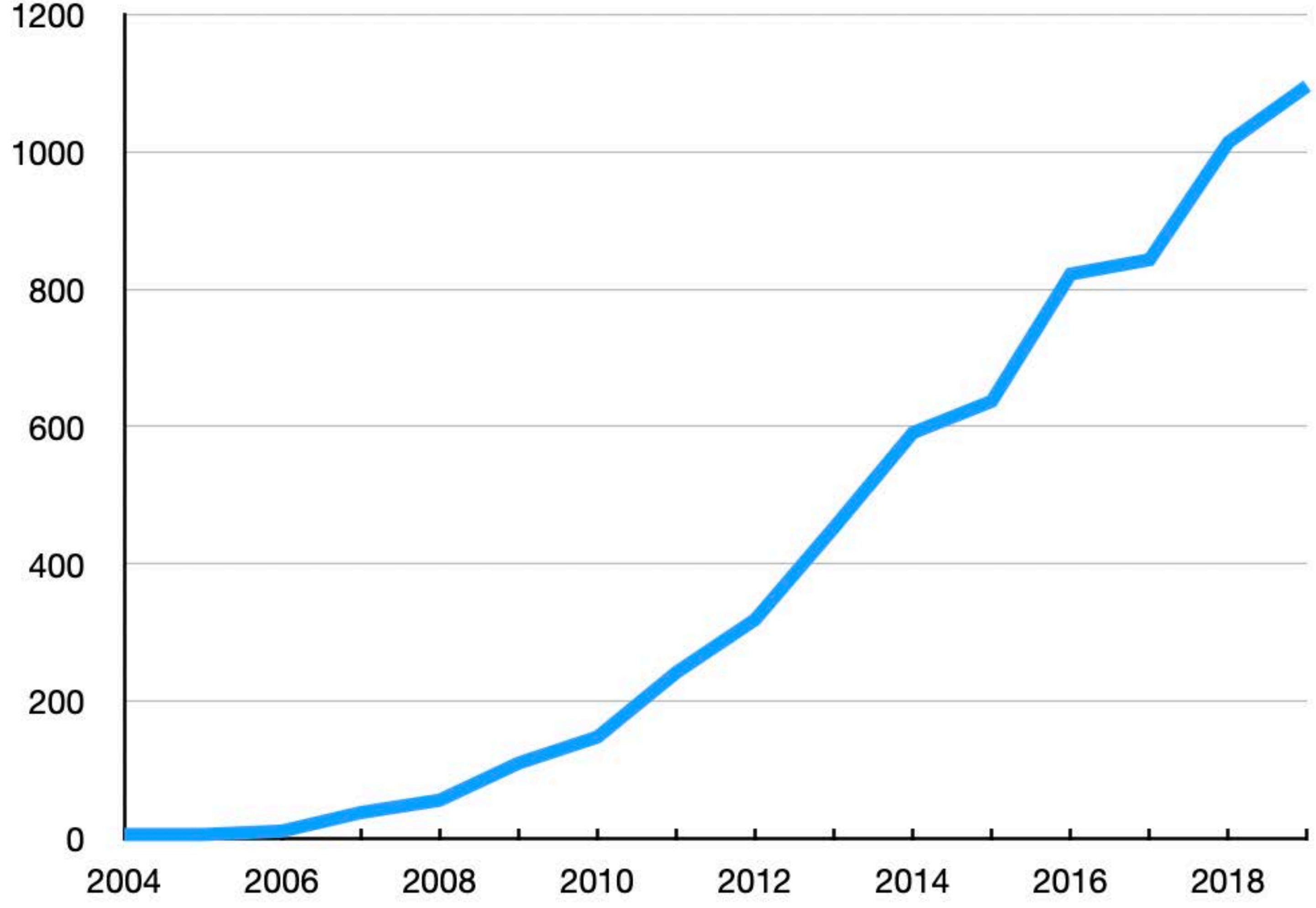
What is community detection?

Networks tend to be organised into modules/ clusters/communities

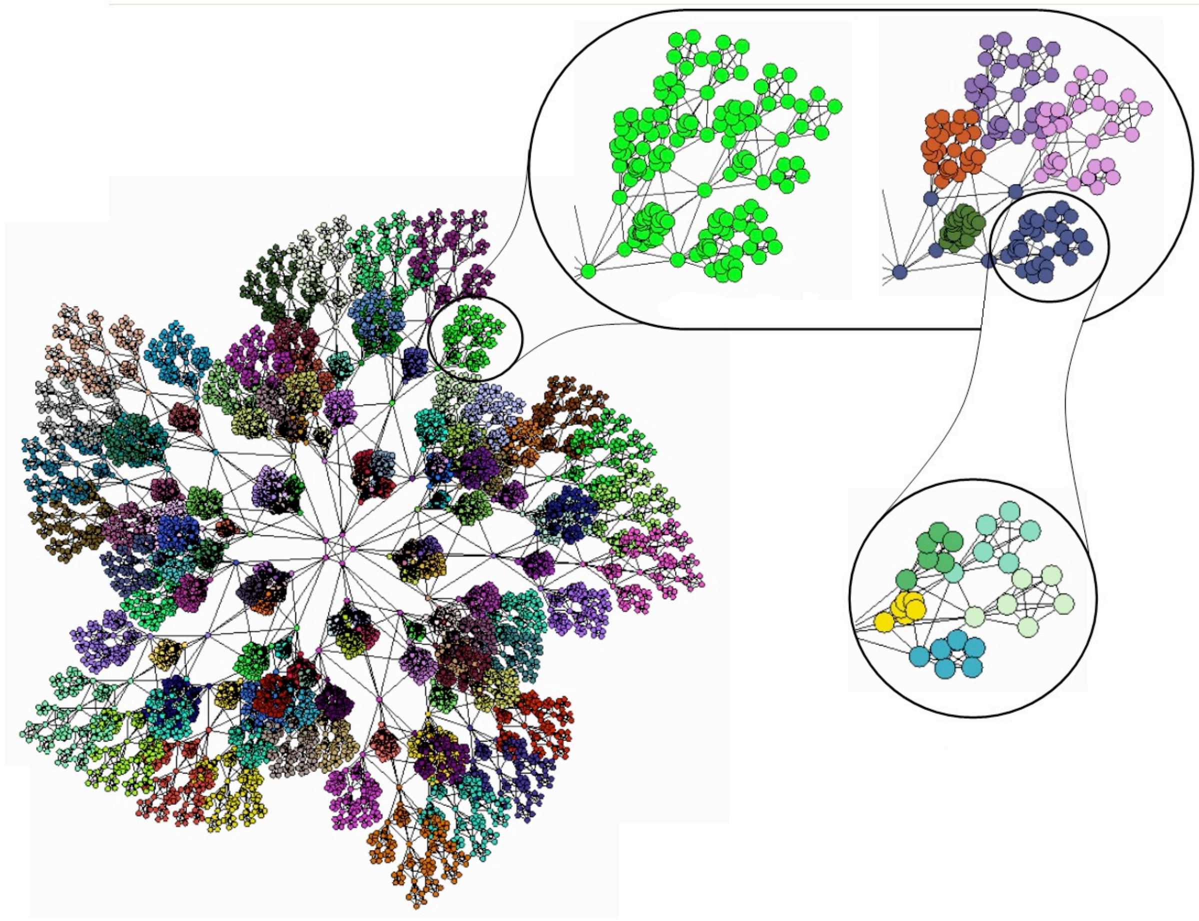
Can we design efficient methods to find the modules? Understand their impact on the behaviour of the system? Find mechanisms that explain their emergence?



“community detection”, SCOPUS, June 2017



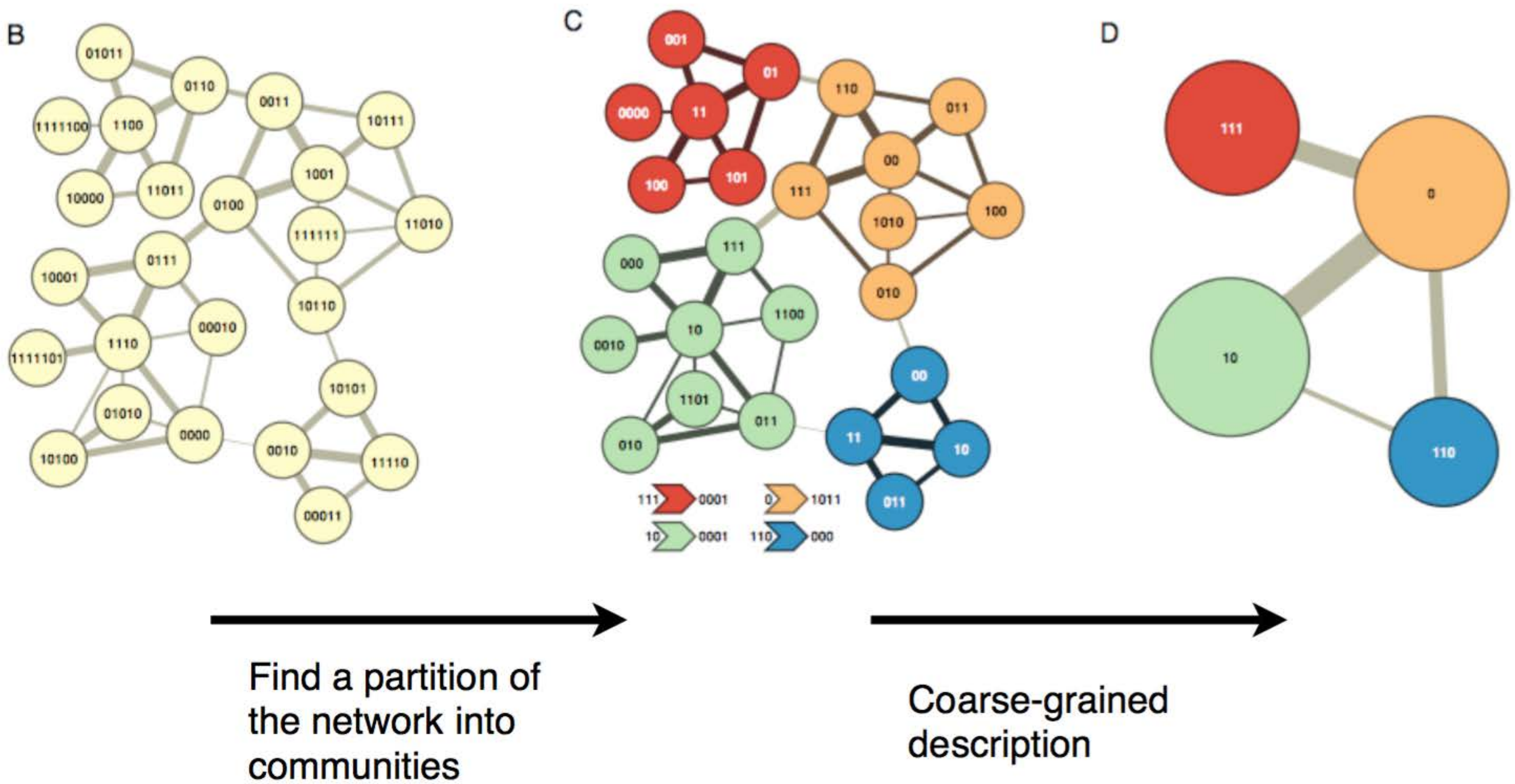
Identify and caracterise the multi-scale structure of networks



Unsupervised clustering in the world of graphs, with applications/connections in/to node classification, embeddings, etc.

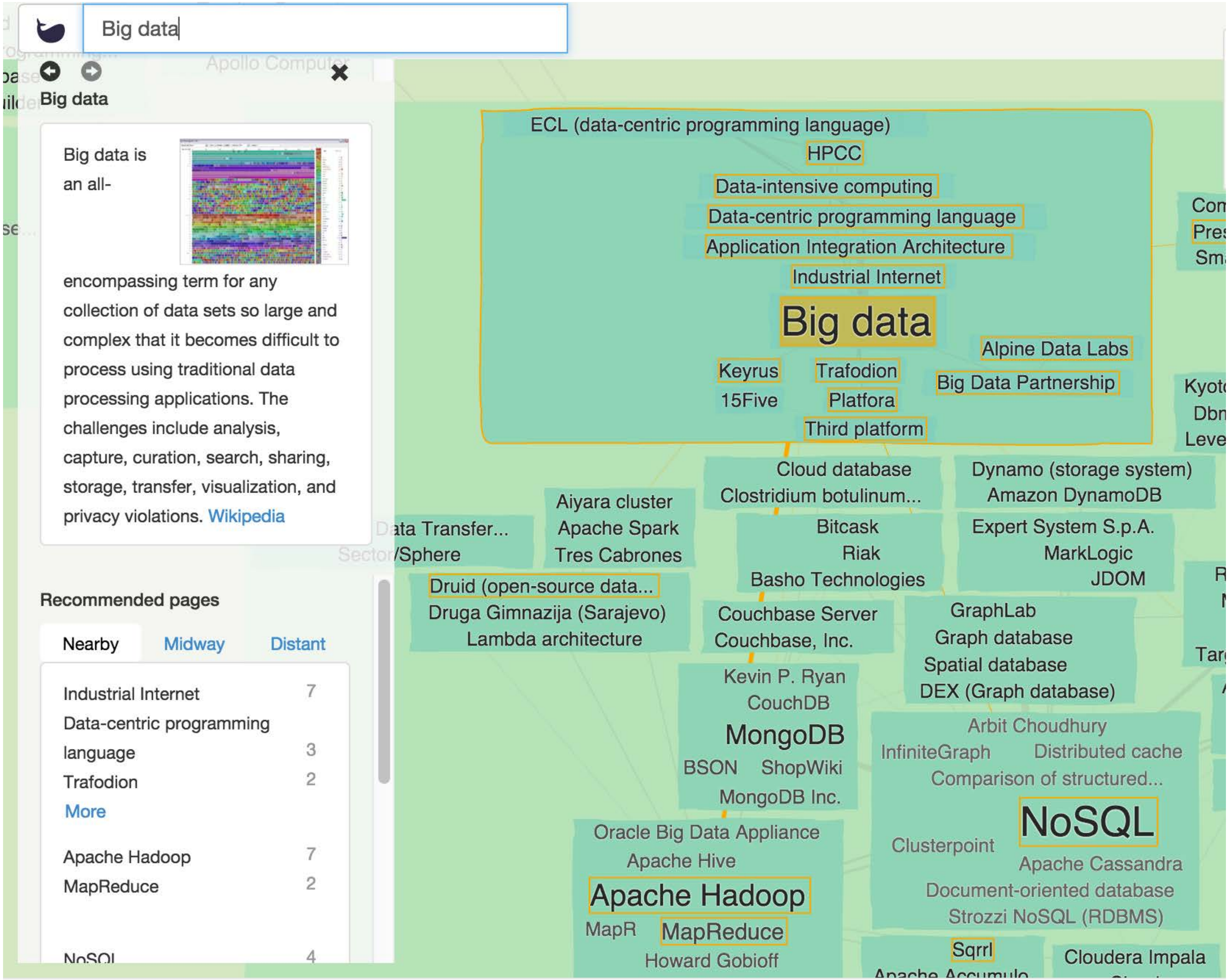
The two key tools of graph analytics: rank and cluster

Uncovering communities/modules helps to change the resolution of the representation and to draw a readable map of the network



Martin Rosvall and Carl T. Bergstrom, PNAS 105, 1118 –1123 (2008)

Uncovering communities/modules helps to change the resolution of the representation and to draw a readable map of the network


Big data
Big data
Big data is an all-encompassing term for any collection of data sets so large and complex that it becomes difficult to process using traditional data processing applications. The challenges include analysis, capture, curation, search, sharing, storage, transfer, visualization, and privacy violations. Wikipedia
Recommended pages
Nearby
Midway
Distant
Industrial Internet 7
Data-centric programming language 3
Trafodion 2
More
Apache Hadoop 7
MapReduce 2
NoSQL 4
ECL (data-centric programming language)
HPCC
Data-intensive computing
Data-centric programming language
Application Integration Architecture
Industrial Internet
Big data
Alpine Data Labs
Keyrus
Trafodion
Big Data Partnership
15Five
Platfora
Third platform
Cloud database
Clostridium botulinum...
Dynamo (storage system)
Amazon DynamoDB
Aiyara cluster
Apache Spark
Tres Cabrones
Bitcask
Riak
Basho Technologies
Expert System S.p.A.
MarkLogic
JDOM
Druid (open-source data...
Druga Gimnazija (Sarajevo)
Lambda architecture
Couchbase Server
Couchbase, Inc.
GraphLab
Graph database
Spatial database
DEX (Graph database)
Kevin P. Ryan
CouchDB
MongoDB
BSON
ShopWiki
MongoDB Inc.
Arbit Choudhury
InfiniteGraph
Distributed cache
Comparison of structured...
NoSQL
Clusterpoint
Apache Cassandra
Document-oriented database
Strozzi NoSQL (RDBMS)
Oracle Big Data Appliance
Apache Hive
Apache Hadoop
MapR
MapReduce
Howard Gobioff
Sqrrl
Cloudera Impala

Why are networks modular?

Generic mechanisms driving the emergence of modularity?

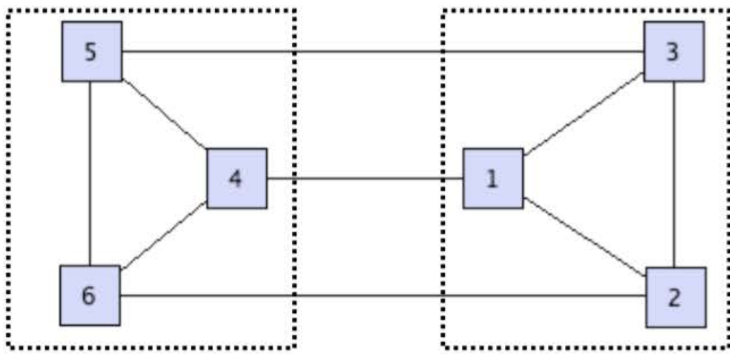
- Simon’s watchmakers: intermediate states facilitates the emergence of complex organisation from elementary subsystems

- Separation of time scales: enhances diversity, locally synchronised states
- Locally dense but globally sparse: advantages of dense structures while minimising the wiring cost
- In social systems, offer the right balance between dense networks (foster trust, facilitate diffusion of complex knowledge), and open networks (small diameter, ensures connectivity, facilitates diffusion of “simple” knowledge)
- Naturally emerges from co-evolution and duplication processes

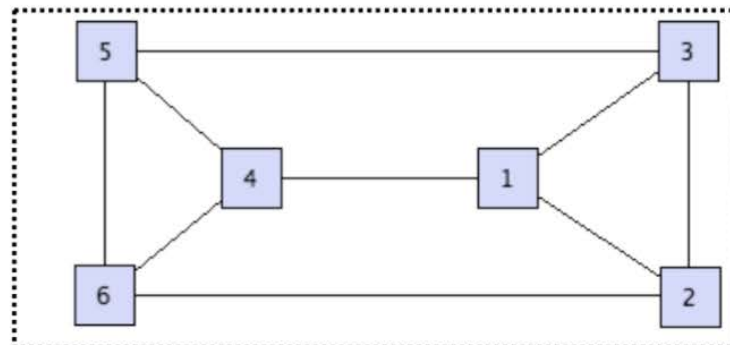
D. Meunier, R. Lambiotte and E.T. Bullmore, “Modular and hierarchical organisation in complex brain networks”, to appear in Frontiers in NeuroScience (2010) - 7 pages

Community detection

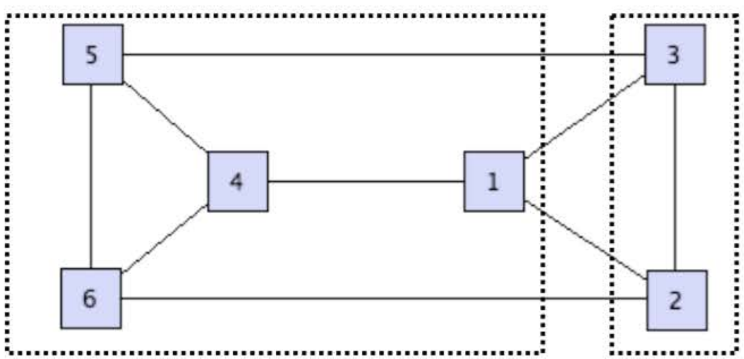
What is the best partition of a network into modules?
How do we rank the quality of partitions of different sizes?



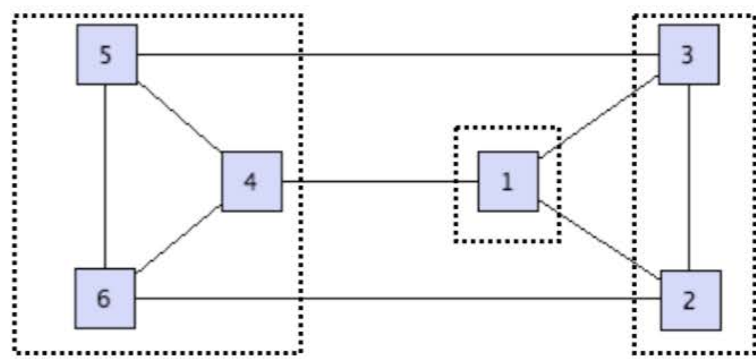
Q1



Q2



Q3



Q4

....

Newman-Girvan Modularity

One of the most popular quality functions for community detection is the so-called Newman–Girvan modularity, denoted by Q.
Let us consider a group of nodes defined by a set **A**. The underlying idea of modularity is to compare the number of links connecting nodes inside **A** with an expectation of this number under a random null model. The choice of null model is, in principle, left to the user. Under the (default) soft-configuration model, the difference between the number of links in community **A** and the expected value of such links is

$$\boxed{\sum_{i,j \in \mathcal{A}} \left(A_{ij} - \frac{k_i k_j}{2m} \right)}$$

Newman-Girvan Modularity

For a partition of a network into multiple communities, Newman-Girvan modularity defines the quality of a partition as a sum of the previous quantity over all communities

$$Q = \frac{1}{2m} \sum_{\mathcal{A}_\alpha \in \mathcal{P}} \sum_{i,j \in \mathcal{A}_\alpha} \left(A_{ij} - \frac{k_i k_j}{2m} \right)$$

It can be rewritten in matrix notations by encoding the partition of the network with a *N X C i*ndicator matrix **H.**

$$Q = \frac{1}{2m} \mathrm{Tr} \left[\boldsymbol{H}^T \left[\boldsymbol{A} - \frac{\boldsymbol{k}\boldsymbol{k}^\top}{2m} \right] \boldsymbol{H} \right]$$

k is the vector of node degrees and Tr denotes the trace of a matrix

Newman-Girvan Modularity

$$Q = \frac{1}{2m}\mathrm{Tr}\left[\boxed{H^T\left[A - \frac{kk^\top}{2m}\right]H}\right]$$

C X C matrix encoding the connections between communities

Inherent to the construction of modularity is the assumption that a network partition with a strong community structure will lead to high values of modularity, in the sense that an unexpectedly large number of edges will be concentrated inside its communities. Modularity optimisation (i.e., finding the partition of a network having the highest value of modularity) has thus been proposed as one way to solve the community detection problem. As modularity optimisation is NP-hard, several heuristics have been proposed.

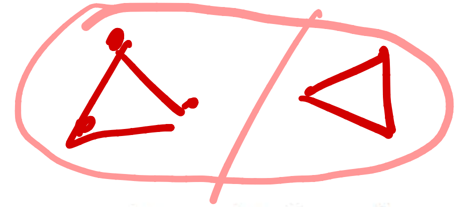
Modularity

Property 1 *A partition where all the vertices are grouped into the same community has a modularity equal to zero. This proves to be simply shown from the definition of the null model* $\frac{k_i k_j}{2m}$ *for which* $\sum_{ij} \frac{k_i k_j}{2m} = 2m$ *and from the expression of modularity in this particular case*

$$\boxed{Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] = 0\ .} \tag{3}$$

This property implies that any partition with a positive modularity is better than this trivial one, but also that it is always possible to find a partition such that $Q \geq 0$.

Modularity



Property 2 *If a partition contains a disconnected community, it is always preferable (in terms of modularity) to split this community into connected communities. Let us consider, for the sake of simplicty, the case of a disconnected community C_1 formed by two connected subgraphs C_{11}, C_{12}. In this case, modularity is given by*

$$
\begin{aligned}
Q &= \frac{1}{2m}\left[\sum_{C\neq C_1}\sum_{i,j\in C}(A_{ij}-\frac{k_ik_j}{2m}) + \sum_{i,j\in C_1}(A_{ij}-\frac{k_ik_j}{2m})\right] \\
&= \frac{1}{2m}\left[\sum_{C\neq C_1}\sum_{i,j\in C}(A_{ij}-\frac{k_ik_j}{2m}) + \sum_{i,j\in C_{11}}(A_{ij}-\frac{k_ik_j}{2m})\right. \qquad (4)\\
&\quad \left. + \sum_{i,j\in C_{12}}(A_{ij}-\frac{k_ik_j}{2m}) + 2\sum_{i\in C_{11}, j\in C_{12}}(A_{ij}-\frac{k_ik_j}{2m})\right].
\end{aligned}
$$

Given that $A_{ij} = 0$ if $i \in C_{11}, j \in C_{12}$, the sum $\sum_{i\in C_{11}, j\in C_{12}}$ is composed uniquely of negative terms and it is thus preferable to split the community into two subcommunities.

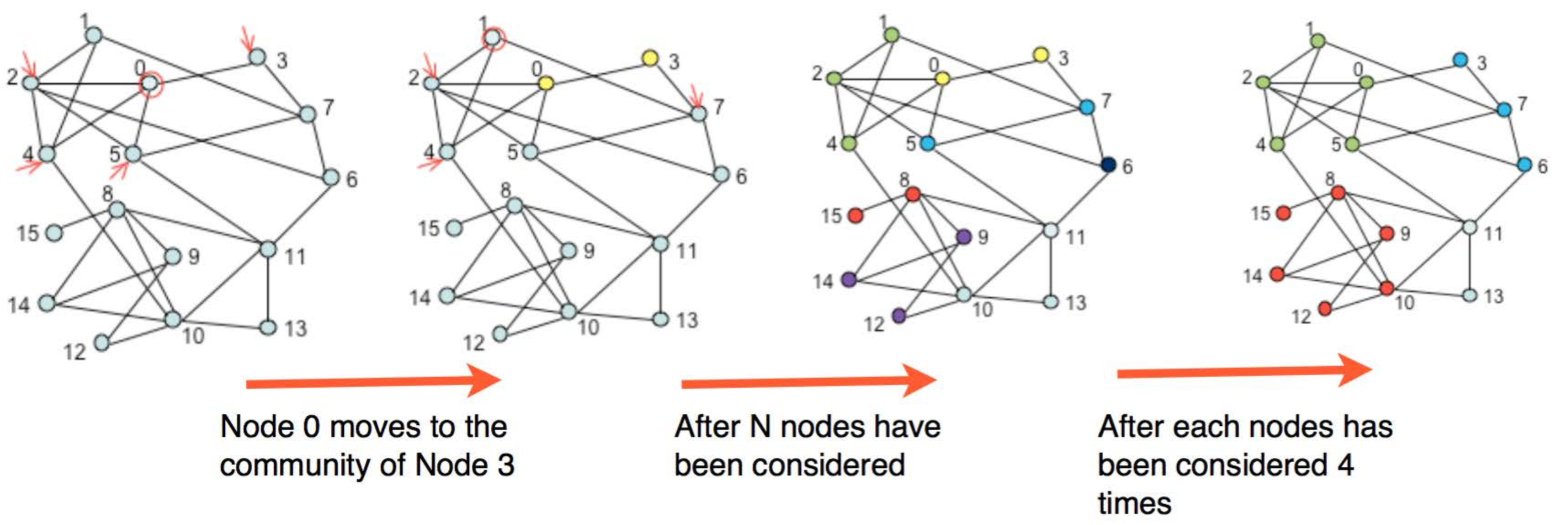
This property implies that any partition made of disconnected communities is sub-optimal and that the optimal partition of a graph is only made of connected communities.

Greedy optimisation

Louvain: multi-scale, agglomerative and greedy

The algorithm is based on two steps that are repeated iteratively. **First phase**: Find a local maximum

1) Give an order to the nodes (0,1,2,3,...., N-1)
2) Initially, each node belongs to its own community (N nodes and N communities)
3) One looks through all the nodes (from 0 to N-1) in an ordered way. The selected node looks among its neighbours and adopt the community of the neighbour for which the increase of modularity is maximum (and positive).
4)This step is performed iteratively until a local maximum of modularity is reached (each node may be considered several times).

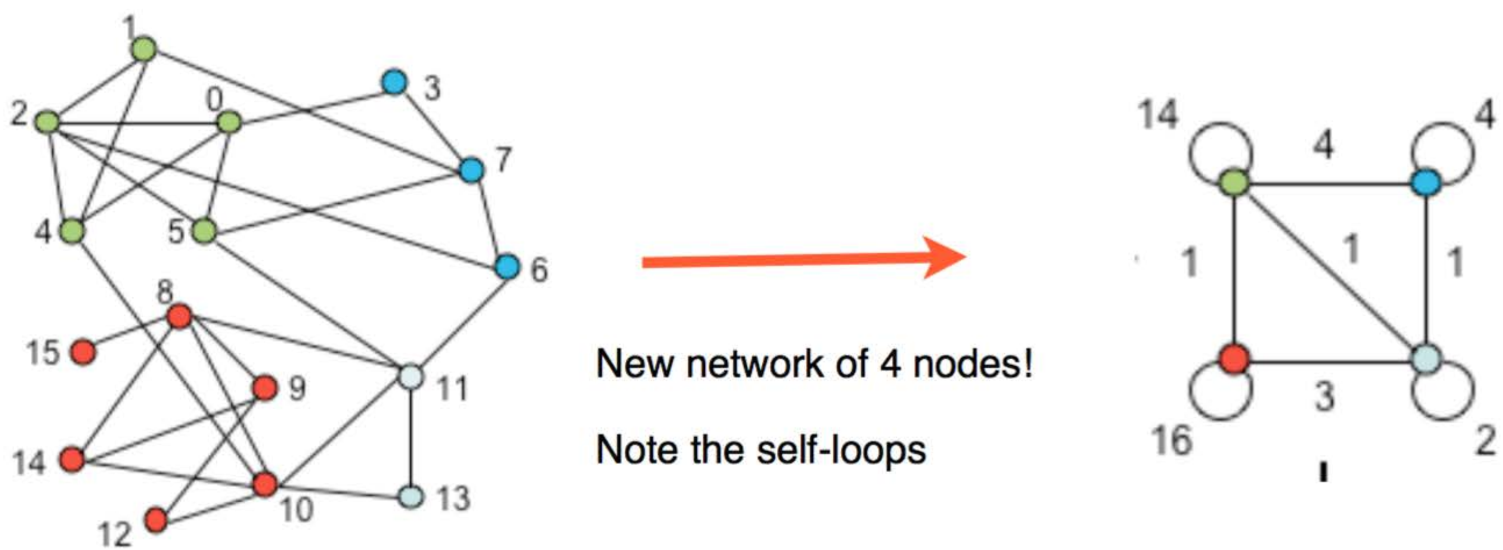


V.D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech., P10008, 2008.

Greedy optimisation

Louvain: multi-scale, agglomerative and greedy

Once a local maximum has been attained, **second phase**:
We build a new network whose nodes are the communities. The weight of the links between communities is the total weight of the links between the nodes of these communities.



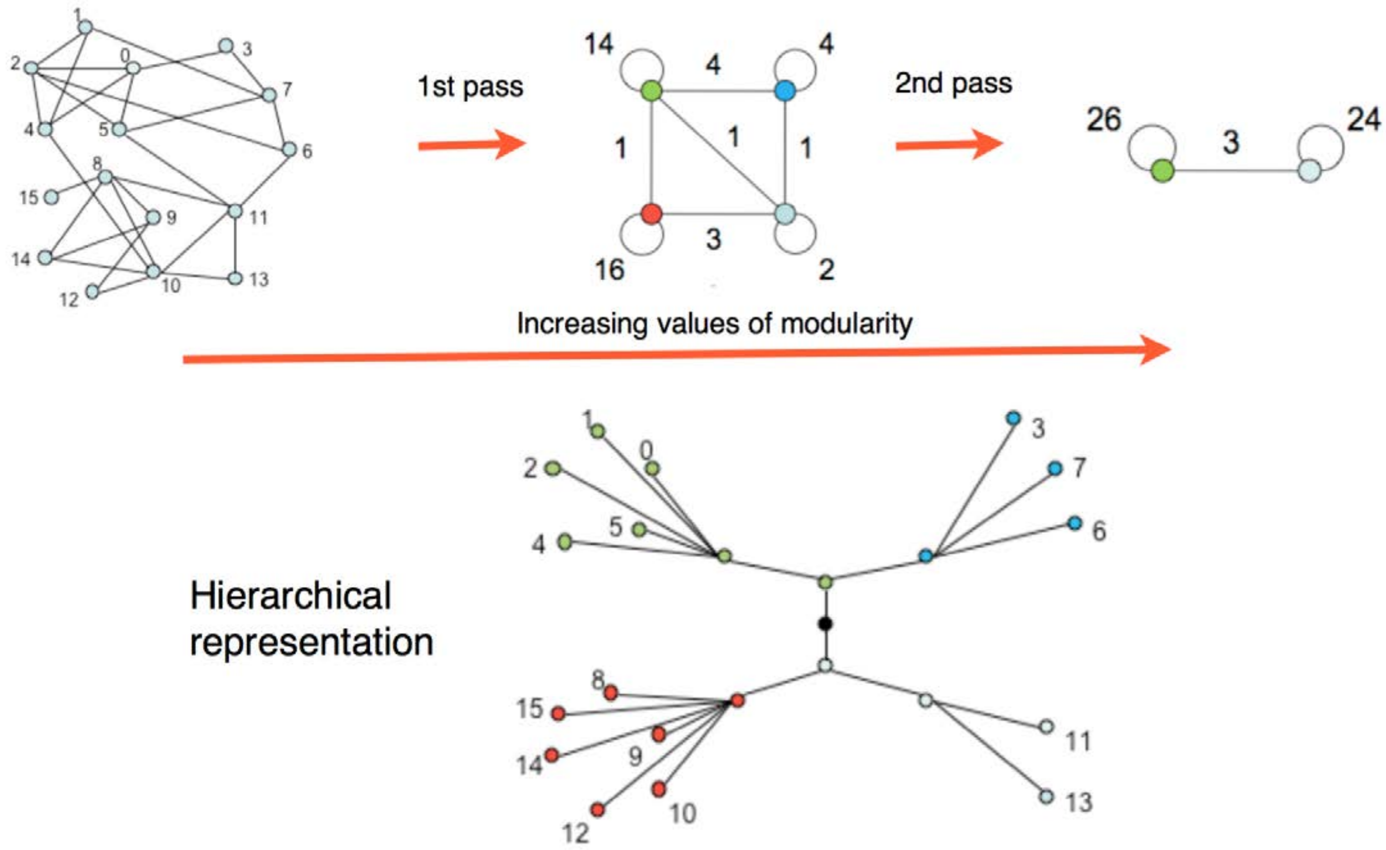
In typical realisations, the number of nodes diminishes drastically at this step.

V.D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech., P10008, 2008.

Greedy optimisation

Louvain: multi-scale, agglomerative and greedy

The two steps are repeated iteratively, thereby leading to a hierarchical decomposition of the network.
Multi-scale optimisation: local search first among neighbours, then among neighbouring communities, etc.



V.D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech., P10008, 2008.

Louvain method

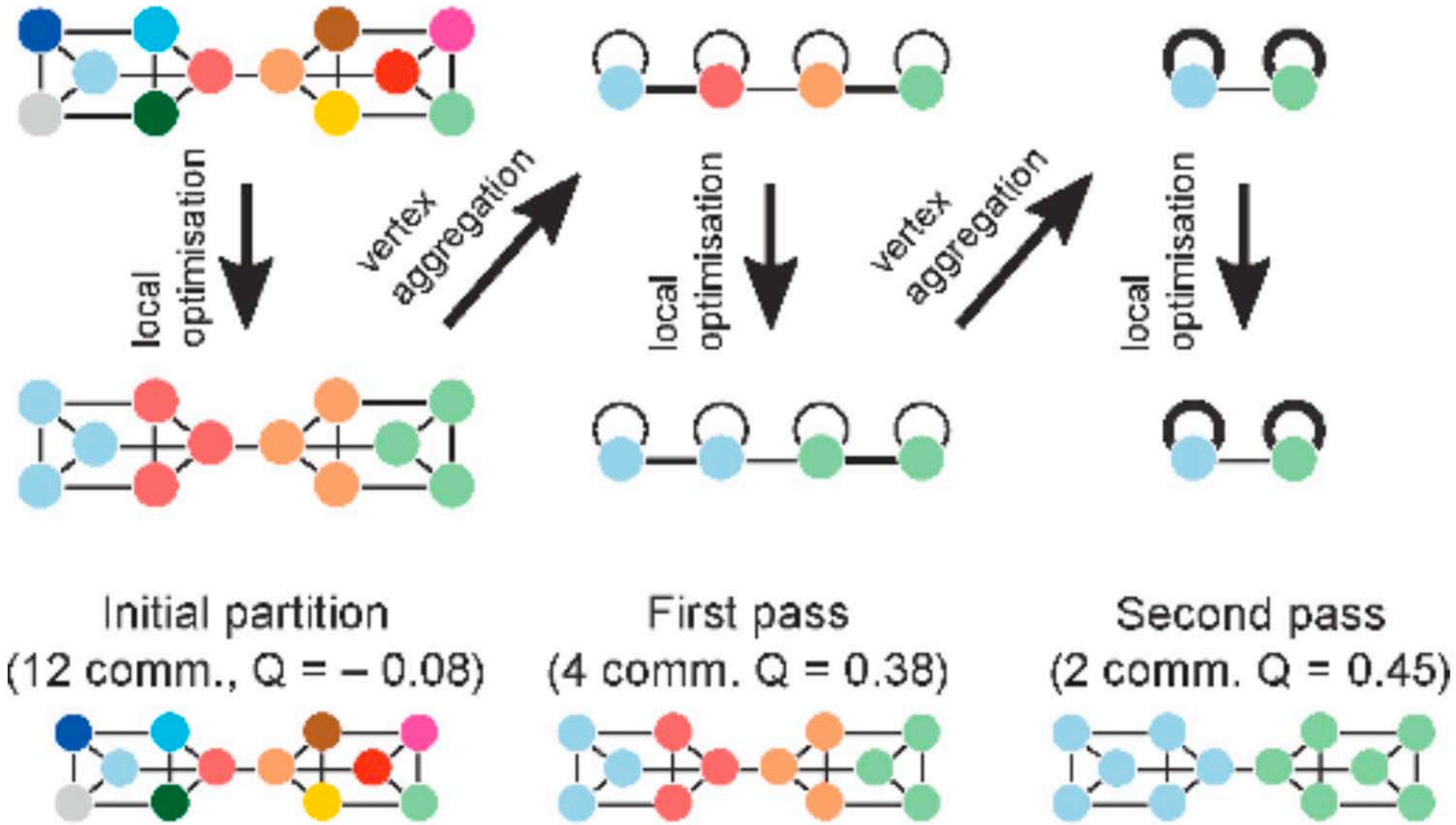


Figure 3 Louvain method for modularity optimisation. The Louvain method consists of repeatedly applied passes over the network until no further increase in modularity is observed. Each pass is split into two phases. The first phase consists in a local optimisation, where each vertex can be moved to the community of its direct neighbours. The second phase aggregates vertices and constructs a meta-graph whose vertices are the communities found after the first phase. Figure adapted from Aynaud et al., (2013).

Greedy optimisation

Louvain: multi-scale, agglomerative and greedy

Very fast: O(N) in practice. The only limitation being the storage of the network in main memory
Good accuracy (among greedy methods)

	Karate	Arxiv	Internet	Web nd.edu	Phone	Web uk-2005	Web WebBase 2001
Nodes/links	34/77	9k/24k	70k/351k	325k/1M	2.04M/5.4M	39M/783M	118M/1B
CNM	.38/0s	.772/3.6s	.692/799s	.927/5034s	-/-	-/-	-/-
PL	.42/0s	.757/3.3s	.729/575s	.895/6666s	-/-	-/-	-/-
WT	.42/0s	.761/0.7s	.667/62s	.898/248s	.553/367s	-/-	-/-
Our algorithm	.42/0s	.813/0s	.781/1s	.935/3s	.76/44s	.979/738s	.984/152mn

V.D. Blondel, J.-L. Guillaume, R. Lambiotte and E. Lefebvre, Fast unfolding of communities in large networks, J. Stat. Mech., P10008, 2008.

Limitations of Modularity (and Louvain)

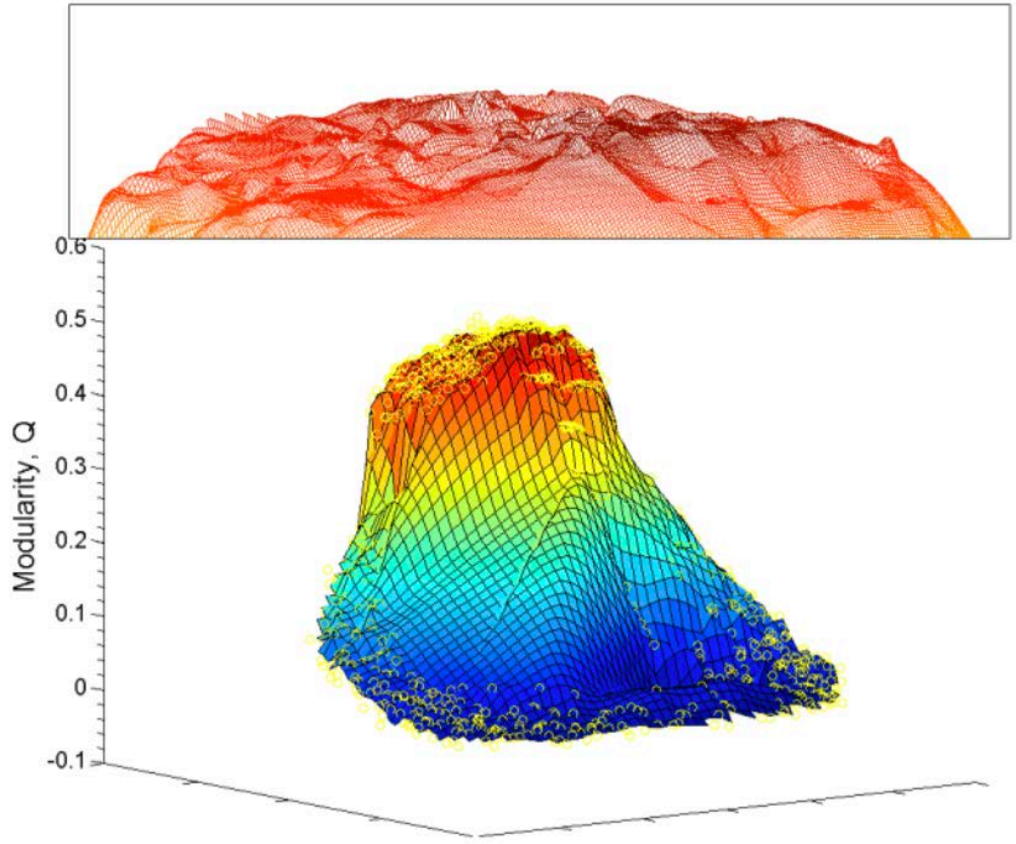
Modularity suffers from a so-called resolution limit which makes it impossible to detect communities of nodes that are smaller than a certain scale. In other words, even if this is not apparent from the definition of modularity at first sight, modularity tends to favour partitions where the communities have a characteristic size depending on the total size of the system.

$$Q = \frac{1}{2m} \sum_{\mathcal{A}_\alpha \in \mathcal{P}} \sum_{i,j \in \mathcal{A}_\alpha} \left(A_{ij} - \boxed{\frac{k_i k_j}{2m}} \right)$$

The null model depends on the total "size" of the network. Optimal communities are size-dependent.

Limitations of Modularity (and Louvain)

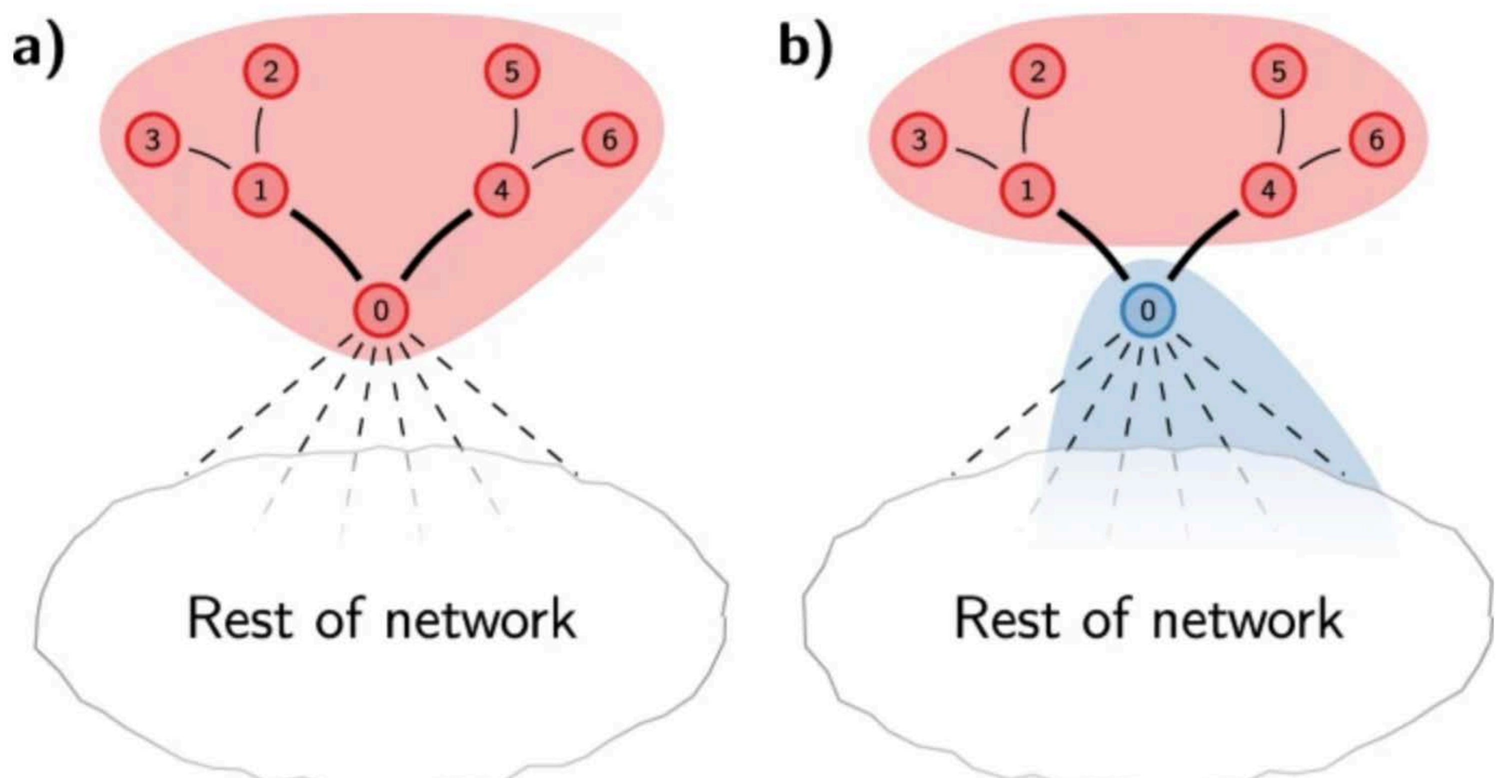
Another limitation of modularity is that its landscape over the space of partitions is usually extremely rugged, with multiple local maxima close to the global optimum, which may limit the interpretability of the approximate solutions found by modularity optimisation.



Limitations of Modularity (and Louvain)

The Louvain algorithm may in certain cases lead to intermediate disconnected communities, which cannot be optimal for modularity. This latter problem can, however, be remedied by adjusting the Louvain algorithm accordingly, e.g. via the Leiden method.

Figure 2



Organisation

H1: Community detection and modularity

H2: Time scale separation and modularity

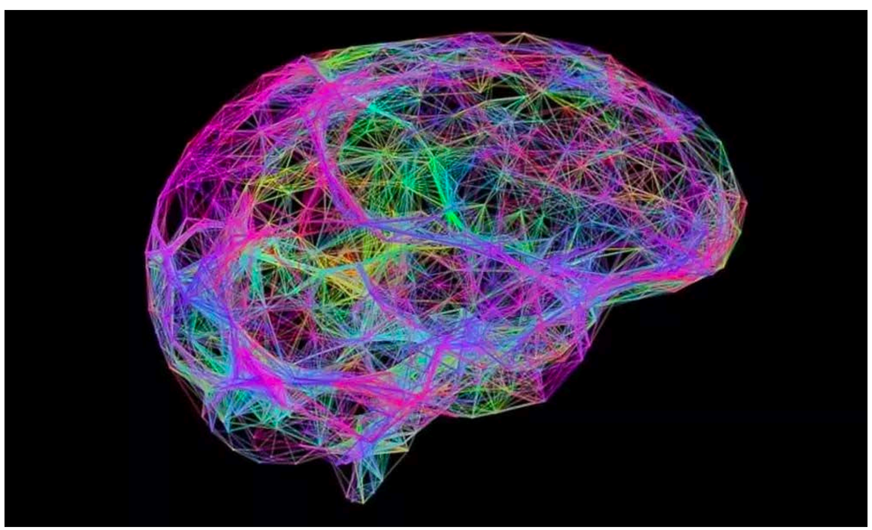
H3: Diffusive dynamics to uncover communities in networks

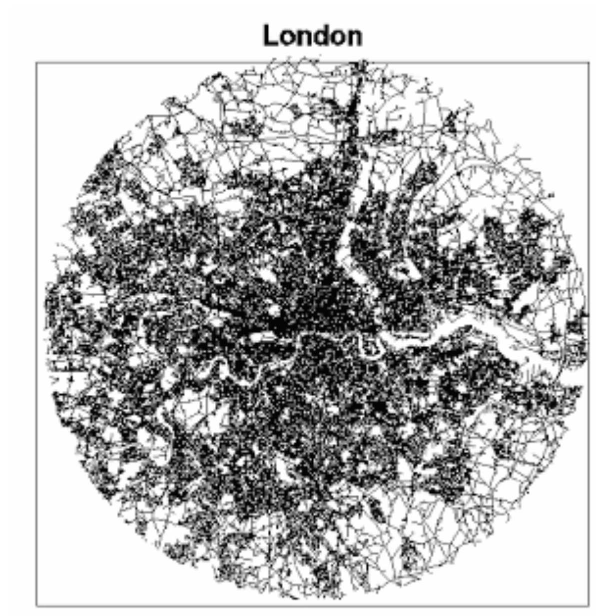
Introduction

Community detection aims to find a coarse-grained description of a system from its structure (e.g. modularity counts edges inside communities)

The function of a system depends on its dynamics.

Important to understand the impact of structure on dynamics, and to find dynamics-based methods to uncover communities.





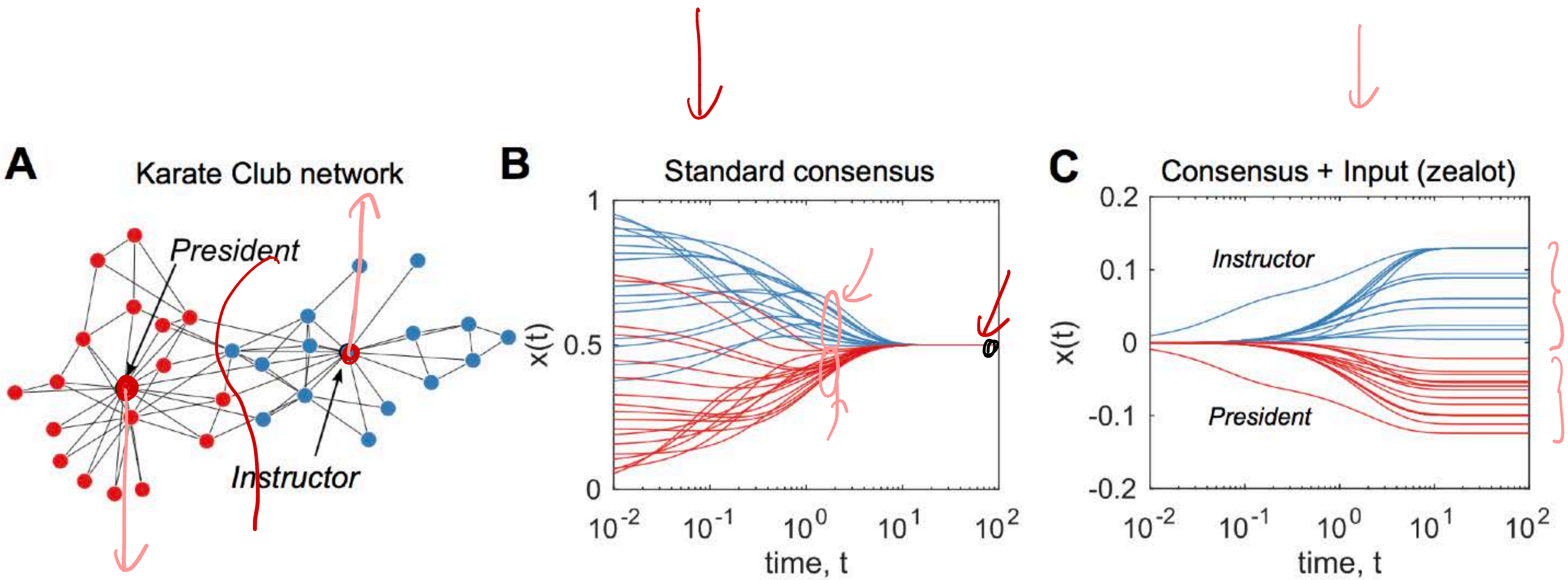


Figure 13.1 Consensus dynamics on the Karate Club network. A The Karate Club network originally analysed by Zachary [8] with nodes coloured according to the split that occurred in the real case. **B** Consensus dynamics on the Karate club network starting from a random initial condition. As time progresses, the states of the individual nodes become more aligned and eventually reach the consensus value equal to the arithmetic average of the initial condition. Note that above the time scale given by the eigenvalue $1/\lambda_2(\mathbf{L}) \approx 1/0.47$, the agents converge into two groups that reflect the observed split before converging to global consensus (see Section 13.3.1). **C** If an external input is applied to the system (see text), the opinion dynamics will in general not converge to a single value but lead to a dispersed set of final opinions, which still reflect the split observed in reality.

Notations

$A \in \mathbb{R}^{n \times n}$ weighted, undirected

$A = A^T$

Weighted degree $\Rightarrow \sum_j A_{ij} = d_i$

$\bar{1}$: $n \times 1$ vector of ones $\Rightarrow \bar{d} = A\bar{1}$

$\mathrm{diag}(\bar{x}) \rightarrow$ matrix X such that $X_{ii} = x_i$ and 0 otherwise.

$D = \mathrm{diag}(\bar{d})$

$$\begin{pmatrix} d_1 & & & 0 \\ & d_2 & & \\ & & \ddots & \\ 0 & & & d_N \end{pmatrix}$$

$$\boxed{L = D - A}$$

Positive semi-definite

$0 \leq \underbrace{\lambda_1}_{\text{graph connected}} \leq \lambda_2 \ldots\ldots \leq \lambda_N$

Consensus dynamics

Opinion formation in a society of individuals
distribution of global functions of sensors and robots

Endow each node i with a state $x_i(t) \in \mathbb{R}$

$$\dot{\bar{x}} = -L\bar{x} \quad \text{(consensus dynamics)}$$

$$\dot{x}_i = \sum_j \left(A_{ij} x_j - \delta_{ij} x_i \right) = \sum_j A_{ij} \left(x_j - x_i \right).$$

$$\bar{x}_0 = \bar{x}(0)$$

Graph connected $\rightarrow$ $\boxed{\text{global consensus}}$

$$x_i \xrightarrow[t \to \infty]{} x_* \qquad x_* = \frac{\bar{1}^T \cdot \bar{x}_0}{n} = \sum_i \frac{x_{0,i}}{n}$$

Time-scale separation

Time-scale separation decouples the system in two regimes and allows to reduce the dimensionality of the dynamics

$$\text{fast} \leftarrow \frac{dx}{dt} = f(x,y)$$

$$\text{slow} \leftarrow \varepsilon^{-1}\frac{dy}{dt} = g(x,y)$$

$\varepsilon \ll 1$

$x(t)$ changes much more rapidly than $y(t)$

$\tau = \varepsilon t$ (slow time variable)

$$\frac{dx}{dt} = f(x,y)$$

$$\frac{dy}{d\tau} = g(x,y)$$

$\Rightarrow$ separation of time scales in the dynamics

At the short term, y has not had the time to evolve

$$\rightarrow \boxed{\frac{dx}{dt} = f(x, y(0))} \leftarrow$$

At the long term, $\frac{dx}{dt} = f(x,y) \rightarrow x^*(y)$

$$\boxed{\frac{dy}{d\tau} = g(x^*(y), y)}$$

Time-scale separation for consensus dynamics

Time-scale separation decouples the system in two regimes and allows to reduce the dimensionality of the dynamics

$$\dot{\bar{x}} = -L\bar{x} \Rightarrow \bar{x} = e^{-Lt} x_0$$

$\Rightarrow$ spectral decomposition of L.

$$L\bar{v}_\alpha = \lambda_\alpha \bar{v}_\alpha$$

$$0 = \lambda_1 < \lambda_2 \dots < \lambda_N$$

$$L = \sum_\alpha \lambda_\alpha \bar{v}_\alpha \bar{v}_\alpha^T$$

$$\bar{x}(t) = \sum_\alpha \underbrace{e^{-\lambda_\alpha t} \bar{v}_\alpha}\, \underbrace{\bar{v}_\alpha^T \cdot \bar{x}_0}$$

N terms

$$\tau_\alpha = 1/\lambda_\alpha$$

$$\underbrace{\{0 \dots \lambda_3\}} << \underbrace{\{\lambda_{3+1} \dots \lambda_N\}}$$

$\tau_0 \angle 1/\lambda_{3+1}$

space spanned by the dominant eigenvectors

vanishingly small

$N \to 3$

Relations between modules (structure) and time-scale separation (spectrum)

For simplicity, let us start with the concrete example of a consensus dynamics $\dot{\boldsymbol{x}} = -\boldsymbol{L}\boldsymbol{x}$ with initial condition $\boldsymbol{x}(0) = \boldsymbol{x}_0$, which takes place on a network composed of C modules with an adjacency matrix of the form

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{A}_1 & & & \\ & \boldsymbol{A}_2 & & \\ & & \ddots & \\ & & & \boldsymbol{A}_C \end{pmatrix} + \boldsymbol{A}_{\text{noise}} =: \boldsymbol{A}_{\text{structure}} + \boldsymbol{A}_{\text{noise}}. \tag{4.4}$$

Each block $\boldsymbol{A}_\alpha$ in the block-diagonal matrix $\boldsymbol{A}_{\text{structure}}$ is supposed to correspond to a densely connected graph, and $\boldsymbol{A}_{\text{noise}}$ is a weak perturbation of this strong assortative modular structure. Let us denote the dimension of each block by n_α, such that $\sum_\alpha n_\alpha = n$.

When there is no noise, the graph is made of C disconnected clusters, and the eigenspace associated to the zero eigenvalue has dimension C. There is a spectral gap with he first non-zero eigenvalue: the zero eigenvalues correspond to modes with no time evolution at all, whereas all other eigenmodes will be associated with an exponentially decaying signal.

When noise is small (the graph is almost disconnected), we use Weyl’s inequality* to show that there is still a spectral gap between the first C eigenvalues and the next N-C values.

*Weyl’s inequalities state that the ordered eigenvalues of the perturbed matrix are close to the eigenvalues of the unperturbed matrix

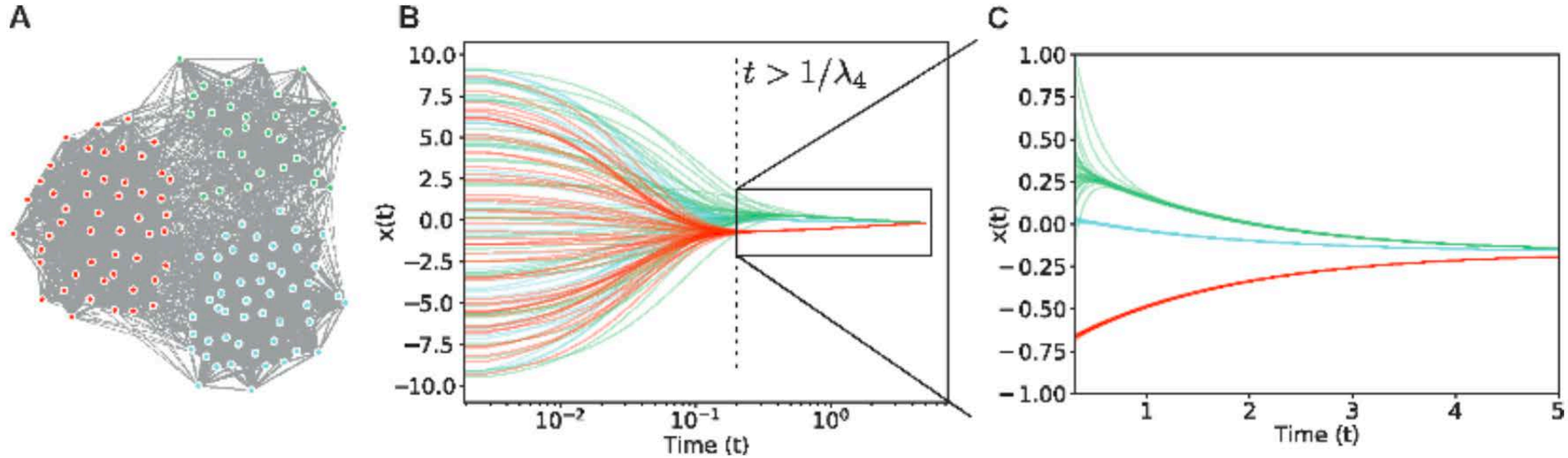
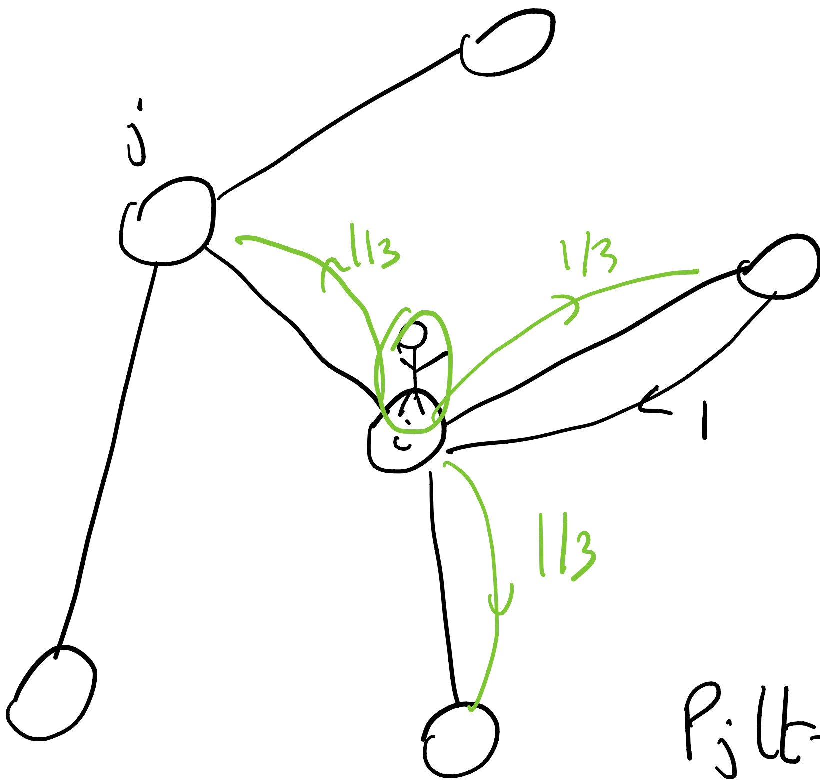


Figure 5 Consensus dynamics on a structured network. A Visualisation of a network with three groups and an adjacency matrix of the form (4.4). **B** When observing a consensus dynamics on this network, there is a clear timescale separation: after $t \approx 1/\lambda_4 = 0.2$, approximate consensus is reached within each group (indicated by color). Eventually global consensus is reached across the network.



$$\bar{P}(t) = [P_1(t) \; \cdots \; P_N(t)]$$

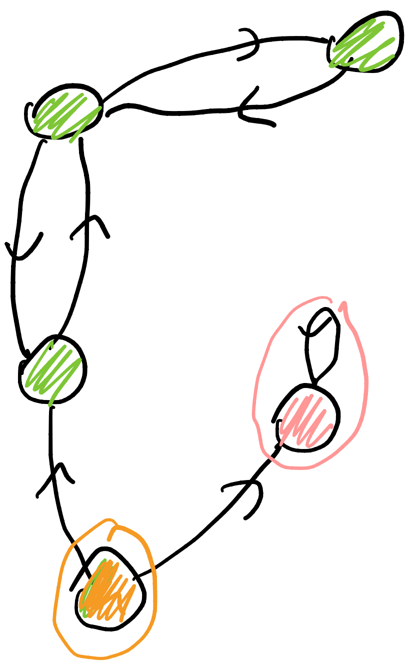
$$A_{ij} \to \underset{i \to j}{T_{ij}} = \frac{A_{ij}}{k_i}$$

$$\sum_j \frac{A_{ij}}{k_i} = \frac{k_i}{k_i} = 1$$

$$P_j(t+1) = \sum_{i=0}^{N} P_i(t)\, T_{ij}$$

$$\bar{P}(t+1) = \bar{P}(t)\, T \leftarrow$$

$$\bar{P}(t) = \bar{P}(0)\, T^t \leftarrow$$



$$\boxed{\sum_j T_{ij} = 1}$$

Absorbing states : $T_{ii} = 1$

$T_{ij} = 0 \quad i \neq j$.

Ergodic set : a set of states such that
- You can go from any i to j, and j to i
- You can not escape the ergodic set

Transient node is a state that is not a member of an ergodic set.

Stationary distributions

$$P^* = P^* T$$

$$P^* = [p_1^* \; \dots \; p_N^*]$$

T

$$T_{ij} = \frac{A_{ij}}{k_i}$$

$$\tilde{A}_{ij} = \frac{A_{ij}}{\sqrt{k_i k_j}} \Rightarrow \tilde{A}_{ij} = \sum_{\ell=1}^{N} \lambda_\ell \bar{u}_\ell \bar{u}_\ell^T$$

$$\langle \bar{u}_\ell, \bar{u}_{\ell'} \rangle = \delta_{\ell\ell'}$$

$$T_{ij} = \sqrt{k_j}\, \tilde{A}_{ij} / \sqrt{k_i} \qquad T = D^{-1/2} \tilde{A} D^{1/2}$$

The same eigenvalues $\Rightarrow$ real

$$u_\ell^L = \left((u_\ell)_1 \sqrt{k_1}, \ldots, (u_\ell)_N \sqrt{k_N} \right)$$

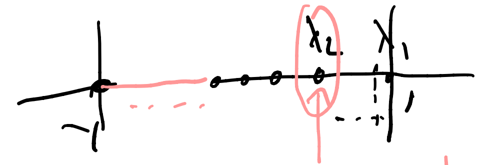
$$u_\ell^R = \left((u_\ell)_1 / \sqrt{k_1}, \ldots, (u_\ell)_N / \sqrt{k_N} \right)$$

$$T^t = \left(D^{-1/2} \tilde{A} D^{1/2} \right)^t = D^{-1/2} \tilde{A} D^{1/2}$$

$$= D^{-1/2} \sum_{\ell=1}^{N} \lambda_\ell^t \bar{u}_\ell \bar{u}_\ell^T D^{1/2} = \sum_{\ell=1}^{N} \lambda_\ell^t \bar{u}_\ell^R \bar{u}_\ell^L$$

$$\bar{p}(t) = p(0) T^t = \sum_{\ell=1}^{N} \lambda_\ell^t u_\ell^L \underbrace{\langle p(0), u_\ell^R \rangle}_{a_\ell(0)}$$

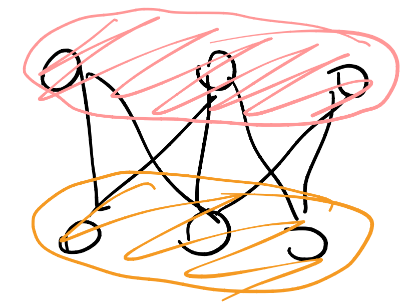
$\lambda_\ell \in [-1, 1]$



$\lambda_\ell = 1$ is unique iff the graph is connected

$\lambda_N = -1$ iff the graph is bipartite.

$\Rightarrow$ the graph is not bipartite



Graph is not bipartite

$t \to \infty$

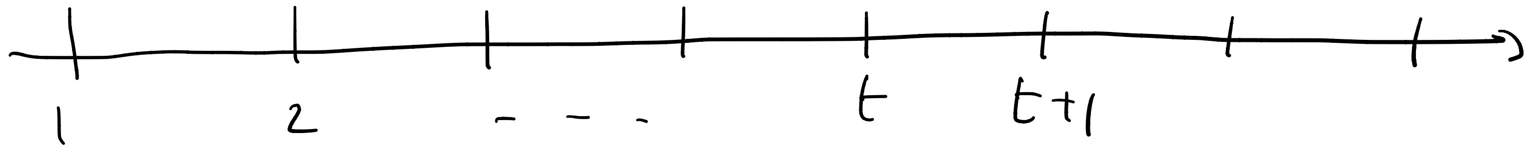
$$p^* = u^L_{max} \underbrace{\langle p(0), u^R_{max} \rangle}_{1}$$

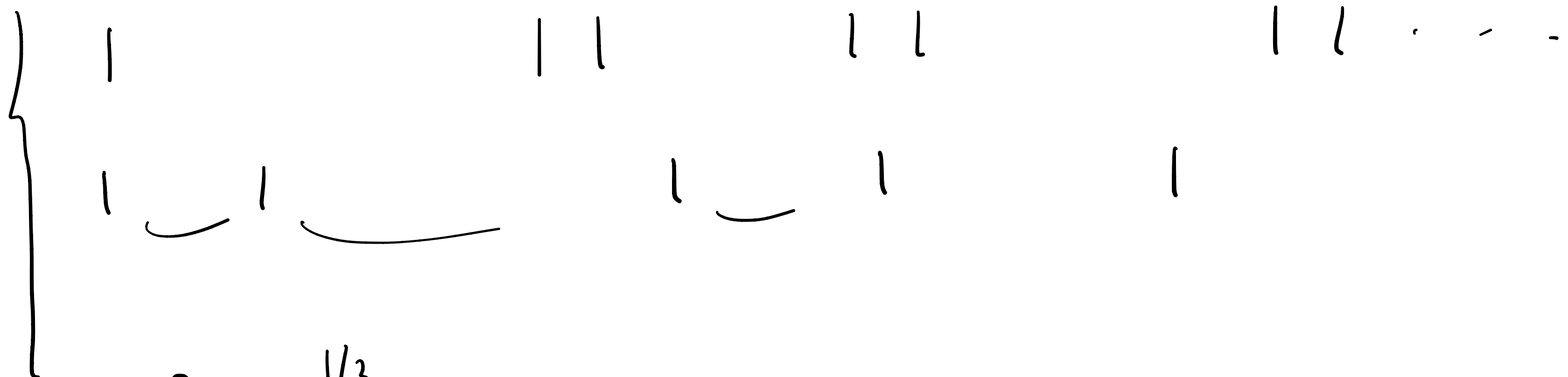
$$u^R_{max} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

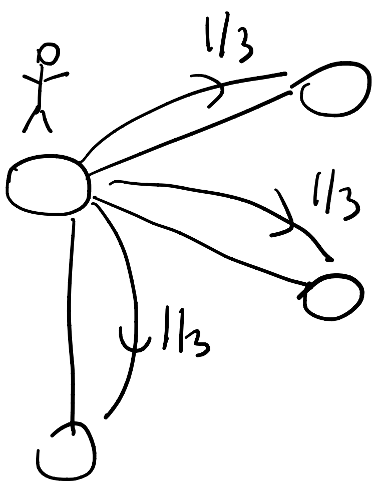
$$P(t) \approx u^L_{max} \langle p(0), u^R_{max} \rangle + \lambda_2^t \, u_2^L \langle P(0), u_2^R \rangle$$

$$|\lambda_\ell|^t \ll |\lambda_2|^t$$

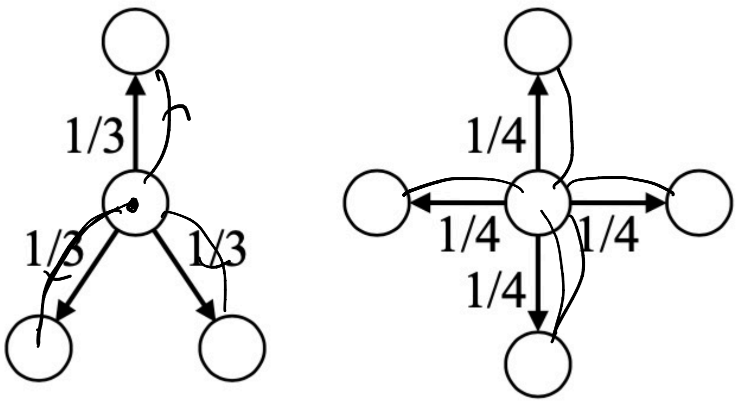
$1 - \lambda_2$ spectral gap.


1
2
t
t+1

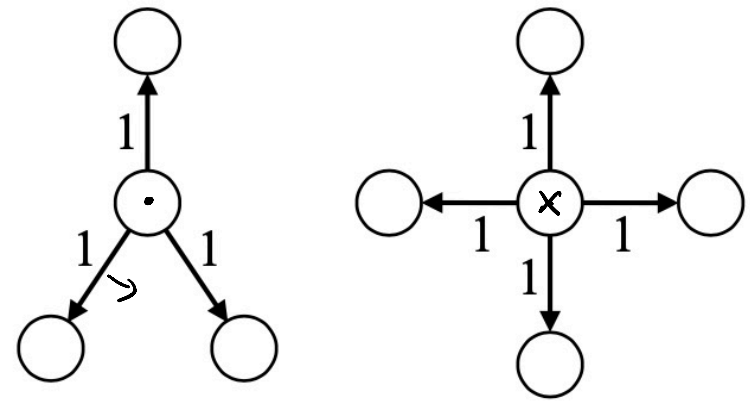



1/3
1/3
1/3

(a) Node-centric CTRW



(b) Edge-centric CTRW



Waits time τ, where τ is a random variable

Independent Poisson process

Inter-event times are exponentially distributed

$\lambda = 1$

Node-centric CTRW

$$\frac{d}{dt}\bar{p}^T(t) = \bar{p}^T(t)\,(-I + T)$$

$$\boxed{= -\bar{p}^T(t)\,\tilde{L}}$$

Random-walk Laplacian

$[-1 \quad 1] \rightarrow [0 \quad 2]$

Stat. state

$$\rightarrow \boxed{p_i^* = \frac{k_i}{2m}}$$

$$\boxed{\frac{d}{dt}\bar{x} = -\tilde{L}\,\bar{x}}$$

Edge-centric CTRW

$$\frac{d}{dt}\bar{p}^T(t) = p^T(t)\,(-D + A)$$

$$\boxed{= -p^T(t)\,L}$$

$$\boxed{\frac{d}{dt}\bar{x} = -L\,\bar{x}}$$

$$\bar{x}^* = \frac{1}{N}\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$$

$$\boxed{p_i^* = \frac{1}{N}}$$

Cheeger inequality

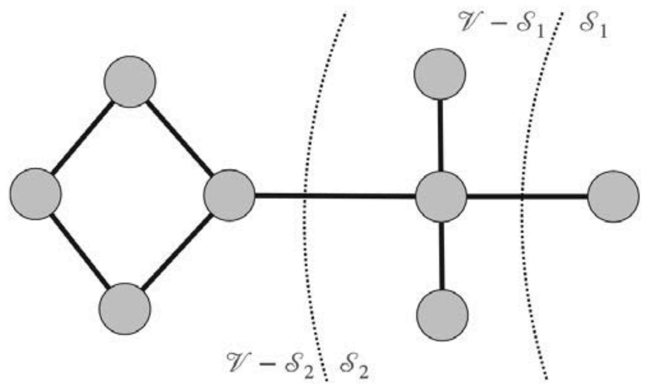


Figure 6 Conductance of a graph. The conductance of the sets of nodes $\mathcal{S}_1$ and $\mathcal{S}_2$, defined by Eq. (4.8), are equal to 1 and 1/7 respectively. Both sets have the same cut size but different volumes. The conductance of $\mathcal{S}_2$ is smaller because it provides a more balanced division of the graph.

$$\phi_{\mathcal{S}} = \left\{ \frac{\sum_{i \in \mathcal{S}, j \notin \mathcal{S}} A_{ij}}{\min\{\mathrm{vol}(\mathcal{S}), \mathrm{vol}(\mathcal{V} - \mathcal{S})\}} \right\} , \tag{4.8}$$

where $\mathrm{vol}(\mathcal{S}) := \sum_{i \in \mathcal{S}} k_i$ is the total connectivity of the set, called the volume of $\mathcal{S}$. The conductance of a graph is then defined as the minimal conductance for all possible node sets: $\phi_{\mathcal{G}} = \min_{\mathcal{S}} \phi_{\mathcal{S}}$. Note that the graph conductance is small if there exist two sets of nodes that are of similar size and have few connections between them. We can now state the Cheeger inequality, which relates the graph conductance to the second smallest eigenvalue of the normalised Laplacian as follows:[12]

$$\frac{\phi_{\mathcal{G}}^2}{2} < \lambda_2 \leq 2\phi_{\mathcal{G}} . \tag{4.9}$$

The Cheeger inequality shows that a small value of the graph conductance is associated to a small spectral gap and hence a comparably slow relaxation of the diffusion dynamics to its stationary state. This means that if the network can be divided into two well-separated node sets with a small cut between them, this bottleneck will slow down diffusion and can thus lead to a separation of timescales.

Organisation

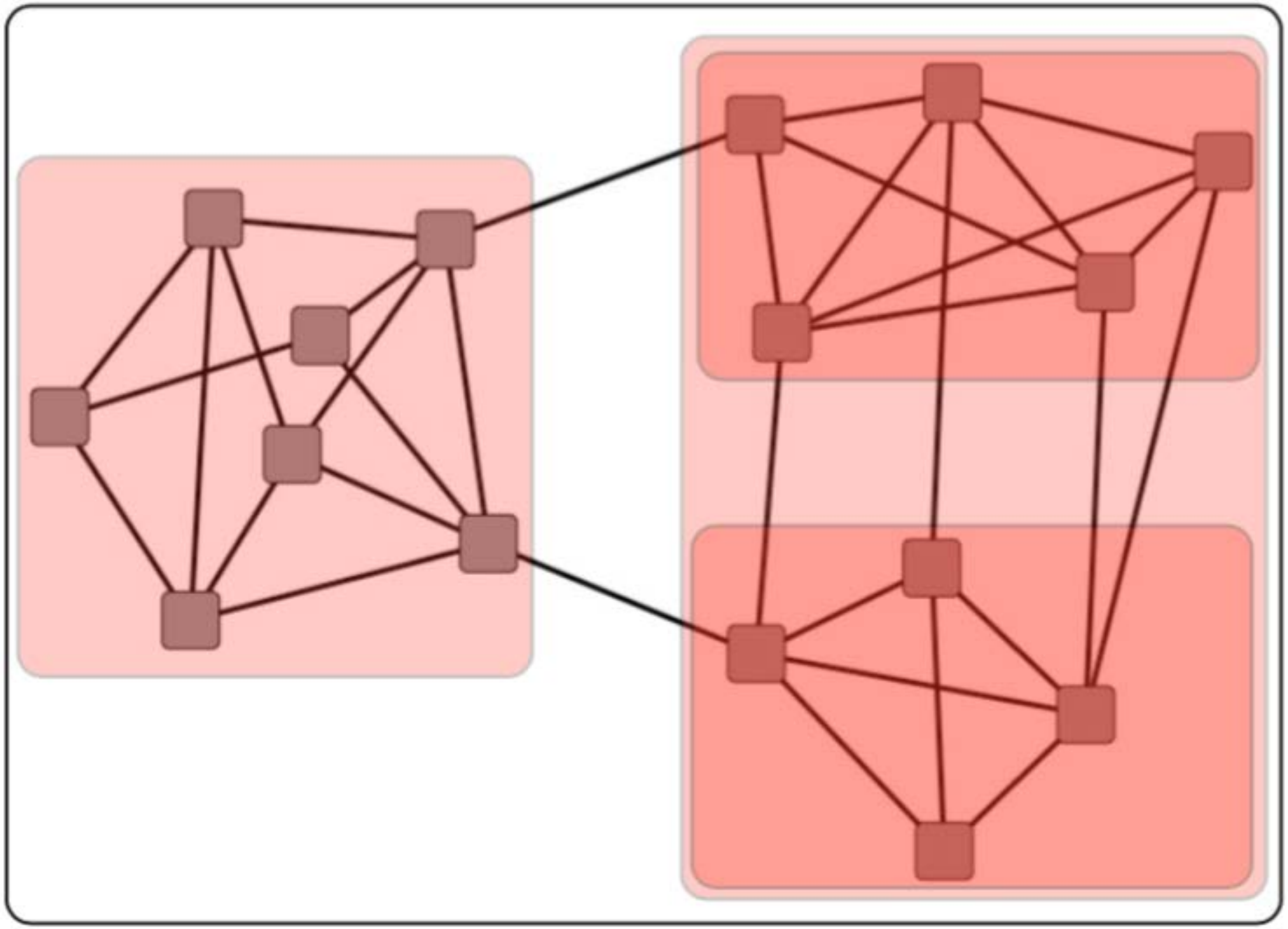
H1: Community detection and modularity

H2: Time scale separation and modularity

H3: Diffusive dynamics to uncover communities in networks

Multi-level modularity

- Resolution limit
- What about sub (or hyper)-communities in a hierarchical network?



Multi-level modularity

Add a resolution parameter!

Reichardt & Bornholdt

$$Q_\gamma = \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \boxed{\gamma P_{ij}}\right]\delta(c_i, c_j)$$

Arenas et al.

$$Q(A_{ij} + rI_{ij})$$

Tuning parameters allow to uncover communities of different sizes
Reichardt & Bornholdt different of Arenas, except in the case of a regular graph where

$$\gamma = 1 + r/\langle k \rangle$$

J. Reichardt and S. Bornholdt, Phys. Rev. E 74, 016110 (2006). Statistical mechanics of community detection
A Arenas, A Fernandez, S Gomez, New J. Phys. 10, 053039 (2008). Analysis of the structure of complex networks at different resolution levels

Multi-level modularity

Add a resolution parameter!

Reichardt & Bornholdt

$$Q_\gamma = \frac{1}{2m}\sum_{i,j}\left[A_{ij} - \gamma P_{ij}\right]\delta(c_i, c_j)$$

Corrected Arenas

$$Q\left(A_{ij} + r\frac{k_i}{\langle k\rangle}\delta_{ij}\right)$$

Preserves the eigenvectors of Laplacian (no A) and has a nice dynamical interpretation

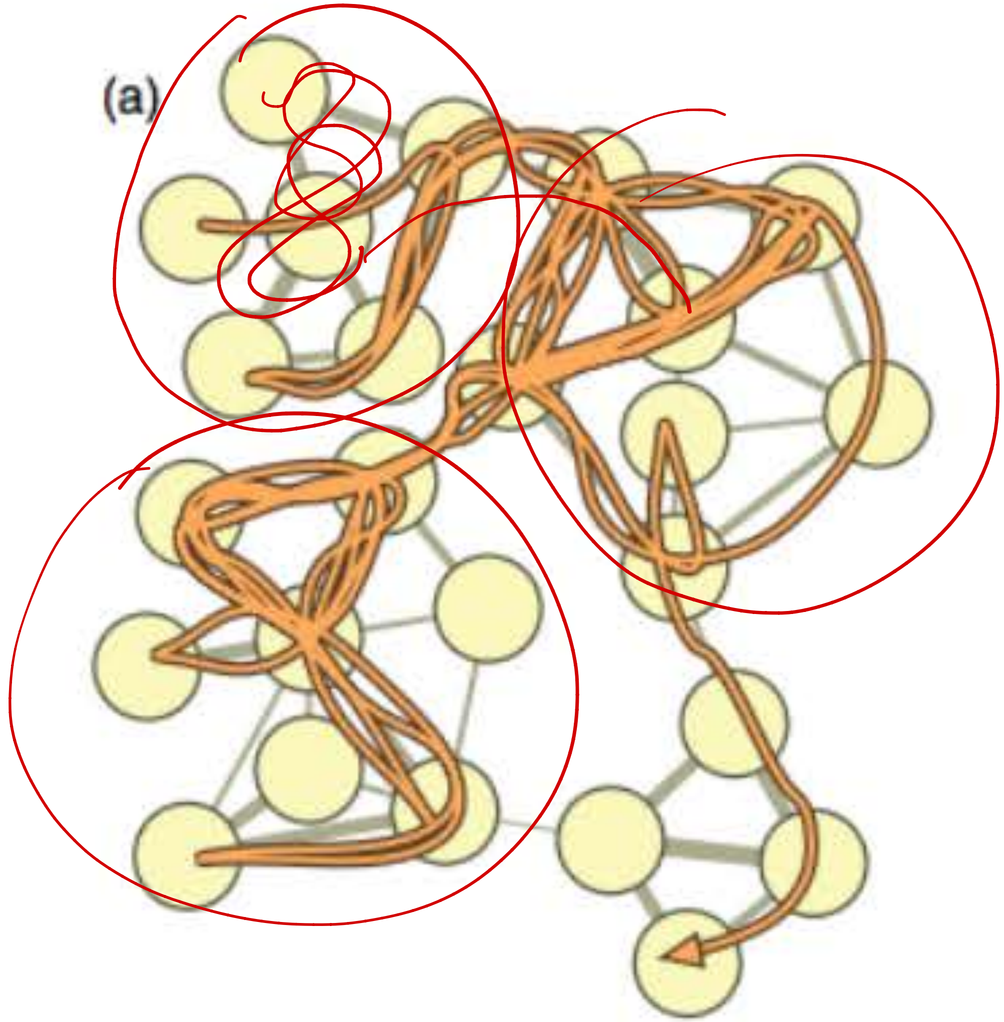
Reichardt & Bornholdt = corrected Arenas for any graph

$$\gamma = 1 + r/\langle k\rangle$$

R. Lambiotte, Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt), 2010 Proceedings of the 8th International Symposium on, 546-553 (2010)

Dynamics as way to uncover communities

time



Say that the system exhibits k slow eigenvectors, our previous discussion implies that the linear system of N equations for the dynamical process can be reduced to a description in a k-dimensional space, spanned by these eigenvectors, in the long time limit.
This classical result from linear dynamical systems theory is particularly helpful to reduce the dimensionality and thus to construct a coarse-graining of a dynamical system.

However, from a network perspective, this solution is not entirely satisfying as the new sets of coordinates are not necessarily concentrated on groups of nodes, and the nodes are the interpretable objects of networks. For this reason, alternative methods have been proposed in order to uncover communities, composed by a definite subset of nodes, that collectively affect dynamics

Markov stability

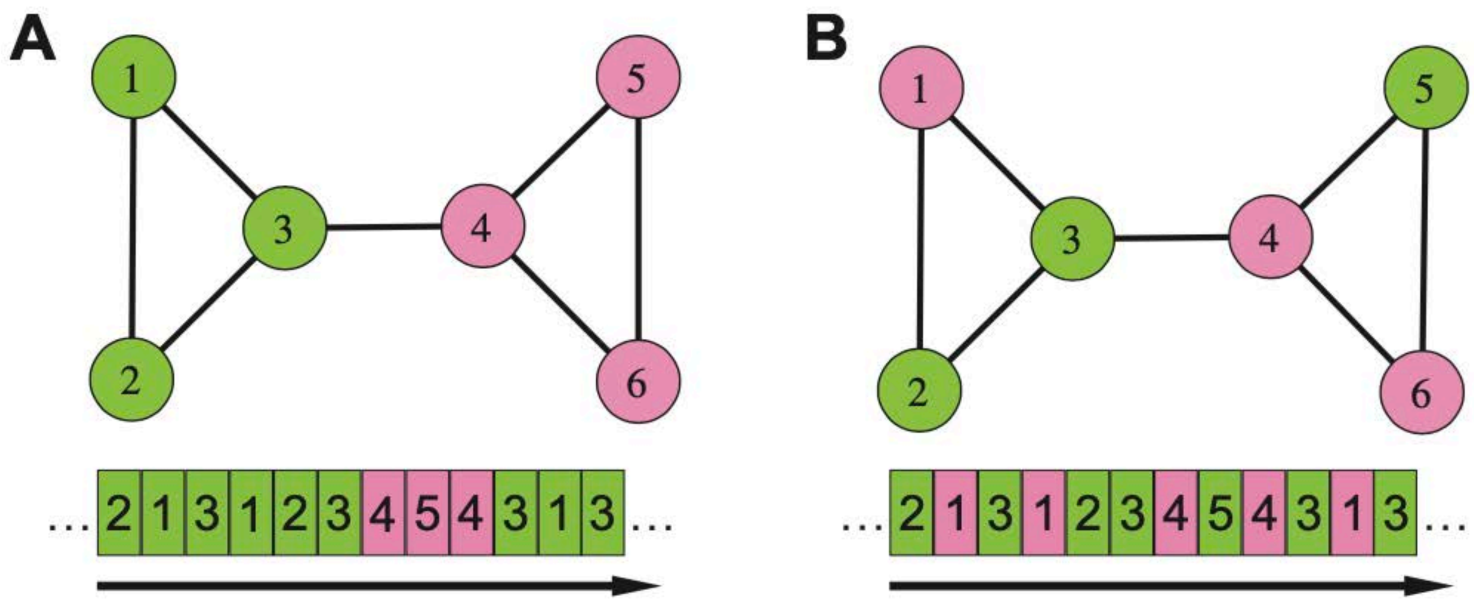


Figure 10 Markov stability and random walks. Given a partition of a graph, here illustrated by two colours, Markov stability is defined by the sequence of communities visited by the random-walk process. Intuitively, for a good partition as in A, the random walker will persist for long times inside a community before escaping it. Markov stability captures the persistence of a random walker at a timescale t via its clustered covariance matrix.

Markov stability

X_α. We now consider the $\boxed{\text{sequence of values } X(t)}$ elicited by a random-walk process on the network, assuming that the random walk has been initialised in its $\boxed{\text{stationary state}}$ at time $t = 0$. The autocovariance of this process evaluated over a period of time t is:

$$\boxed{\operatorname{cov}[X(0)X(t)] = \mathbb{E}[X(0)X(t)] - \mathbb{E}[X(0)]\mathbb{E}[X(t)],} \tag{6.3}$$

where $\mathbb{E}[X(t)]$ is the expectation of the random variable $X(t)$. For a discrete-time random walk, this autocovariance is given by

$$\operatorname{cov}[X(0)X(t)] = \boldsymbol{X}^\top \boldsymbol{R}(t, \boldsymbol{H})\boldsymbol{X}, \tag{6.4}$$

where $\boldsymbol{X}$ is the $1 \times C$ column vector of labels assigned to the C communities and where

$$\boxed{\boldsymbol{R}(t, \boldsymbol{H}) = \boldsymbol{H}^\top \left[\boldsymbol{\Pi} \boldsymbol{T}^t - \boldsymbol{\pi}\boldsymbol{\pi}^\top\right] \boldsymbol{H}} \tag{6.5}$$

is by definition the $C \times C$ clustered covariance matrix. In this last expression, $\boldsymbol{\Pi} = \operatorname{diag}(\boldsymbol{\pi})$ is a diagonal matrix encoding the stationary distribution of the random walk ($\boldsymbol{\pi}^\top = \boldsymbol{\pi}^\top \boldsymbol{T}$).

Markov stability

Observe that the clustered autocovariance matrix $\boldsymbol{R}(t, \boldsymbol{H})$ does not depend on the arbitrary values X_α used to encode the communities. By construction, $(\boldsymbol{\Pi T}^t)_{ij}$ measures the flow of probability from node i to node j in t steps, starting from the stationary distribution of the random walk. Due to the multiplication by the indicator matrices, the term $[\boldsymbol{H}^\top \boldsymbol{\Pi T}^t \boldsymbol{H}]_{\alpha\beta}$ thus measures the flow of probability between any two communities $\mathcal{A}_\alpha$ and $\mathcal{A}_\beta$ over time t. Moreover, as we have assumed that the dynamics is ergodic, the probability to arrive on node j becomes independent of its initial state in the long time limit:

$$\lim_{t\to\infty} (\boldsymbol{\Pi T}^t) = \boldsymbol{\pi}\boldsymbol{\pi}^\top. \tag{6.6}$$

Hence, the second term in the clustered covariance, $\boldsymbol{H}^\top \boldsymbol{\pi}\boldsymbol{\pi}^\top \boldsymbol{H}$, describes the flow of probability between two communities as $t \to \infty$. Note that this also implies that all the elements of $\boldsymbol{R}(t, \boldsymbol{H})$ will converge to zero as $t \to \infty$, irrespectively of the partition considered.

Markov stability

In general, the (α, β) entry of the $\mathcal{C} \times \mathcal{C}$ matrix $\boldsymbol{R}(t, \boldsymbol{H})$ describes the probability that a random walker will be at community $\mathcal{A}_\alpha$ at time zero and community $\mathcal{A}_\beta$ at time t, minus the probability of these events happening by chance at the stationary state. Intuitively, there is a strong assortative community structure over a timescale t, if the probability flows are contained within the communities, hence concentrating high values on the diagonal of $\boldsymbol{R}(t, \boldsymbol{H})$. Accordingly, the Markov stability for discrete-time random walks is defined via the trace of the clustered autocovariance matrix (Delvenne et al., 2013, 2010):

Markov Stability at time t: $\rightarrow$ $\mathrm{Tr}\left[\boldsymbol{R}(t, \boldsymbol{H})\right]$

Markov stability versus Modularity

Let us consider a random walk on an undirected network:

$$\boxed{p_{i;n+1} = \sum_j \frac{A_{ij}}{k_j} p_{j;n}} \xrightarrow{\text{equilibrium}} \boxed{p_i^* = k_i/2m}$$

$$R(1) = \sum_{i,j} \left[\frac{A_{ij}}{k_j} \frac{k_j}{2m} - \frac{k_i k_j}{(2m)^2} \right] \delta(c_i, c_j)$$

Probability that a walker is in the same community initially and at time t=1

Same probability for independent walkers

$$\boxed{R(1) = Q = \frac{1}{2m} \sum_{i,j} \left[A_{ij} - \frac{k_i k_j}{2m} \right] \delta(c_i, c_j)}$$

Markov stability versus Modularity

Let us consider a random walk on an **directed** network:

$$p_{i;n+1} = \sum_j \frac{A_{ij}}{k_j^{out}} p_{j;n}$$

equilibrium

$$p_i^* = \pi_i$$

$$\boxed{R(1) = \sum_{i,j} \left[\frac{A_{ij}}{k_j^{out}} \pi_j - \pi_i \pi_j \right] \delta(c_i, c_j) \neq Q}$$

$$\sum_{ij} \left[A_{ij} - \frac{k_i^{in} k_j^{out}}{m} \right] \delta(c_i, c_j)$$

Counting versus flows

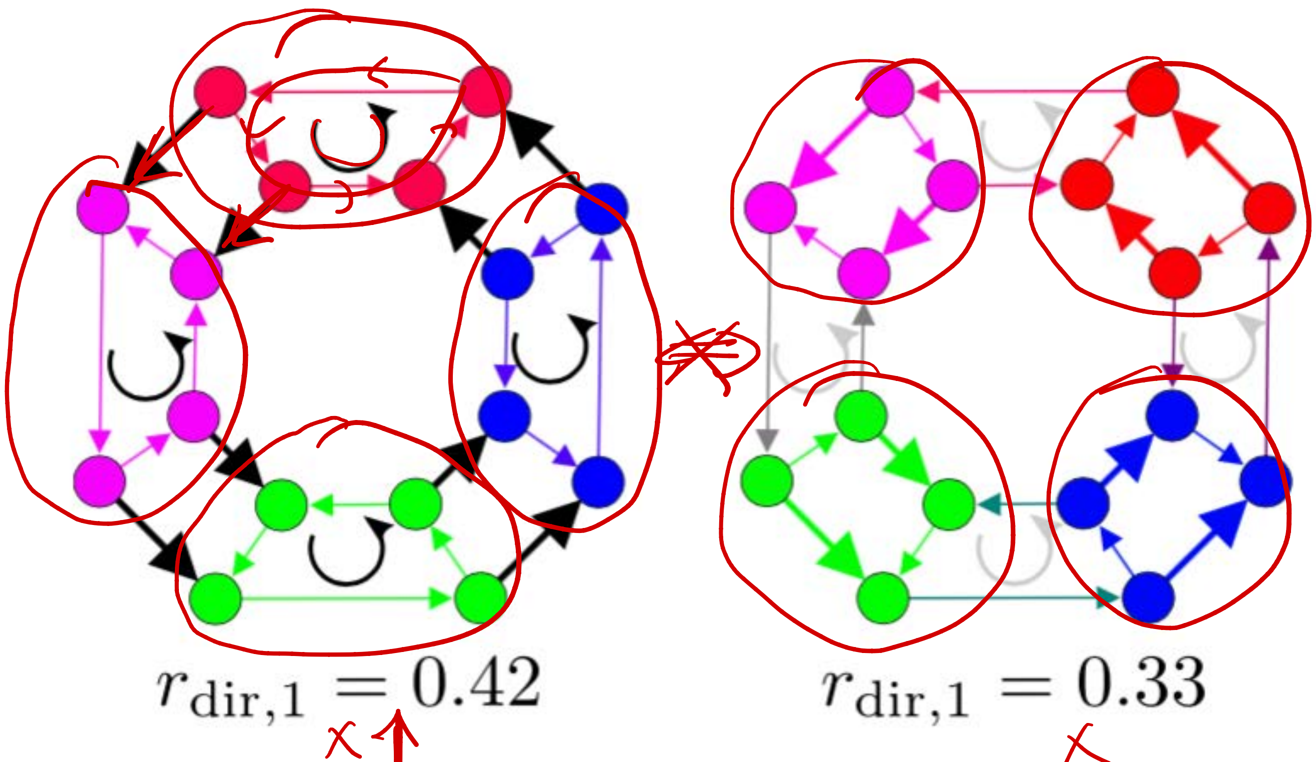
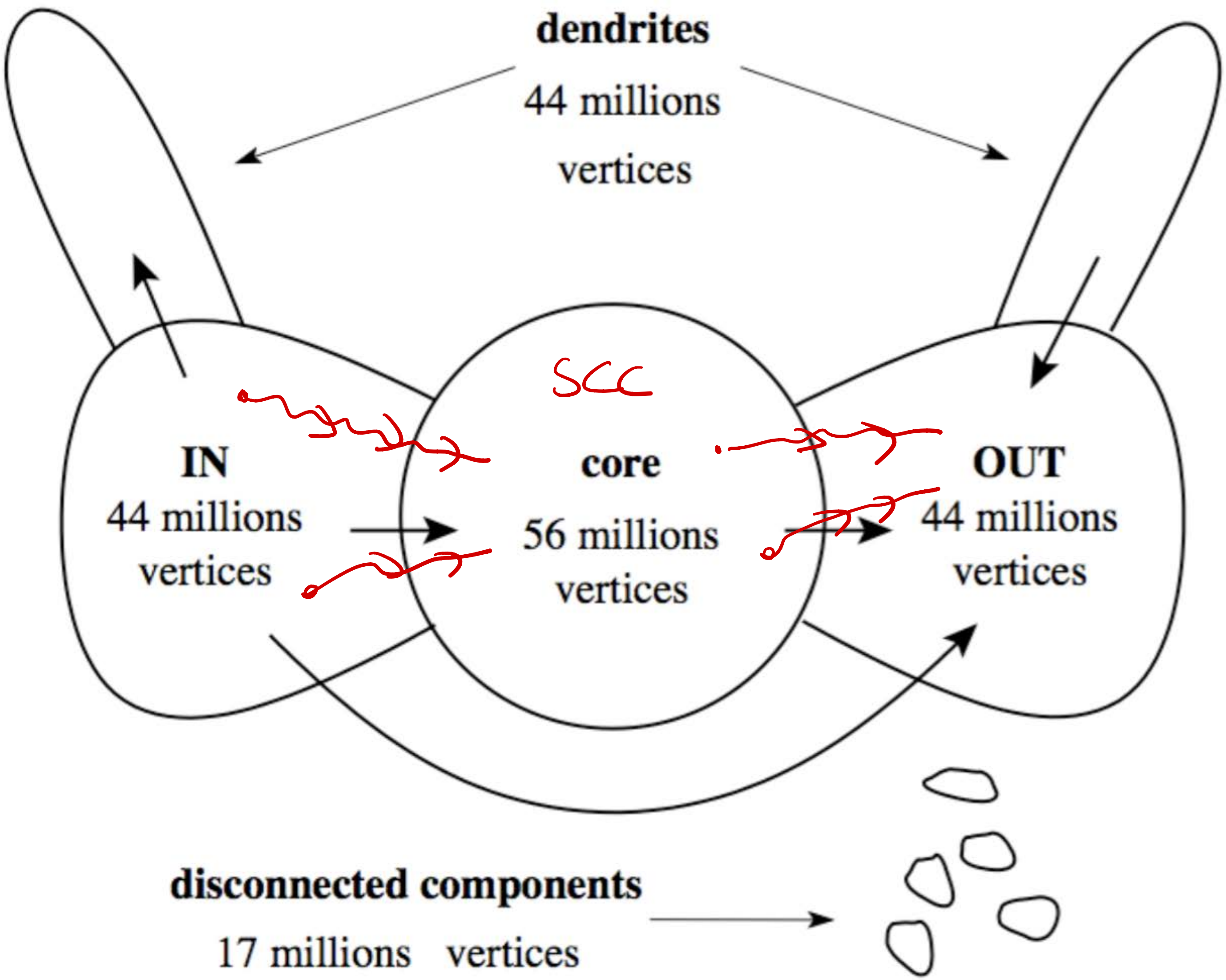


Fig. 4. **Directed Markov Stability versus extensions of modularity.** In this toy network [16], the weight of the bold links is twice the weight of the other links. The partition on the left (indicated by different colors) optimizes directed Markov Stability (34), which intrinsically contains the pagerank as a null model. The partition on the right instead optimizes an extension of modularity based on in- and out-degrees [64], [65]. Hence directed Markov Stability produces flow communities, whereas the extension of modularity ignores the effect of flows.

Counting versus flows



Markov stability versus Modularity

Let us consider a random walk on an **directed** network:

$$p_{i;n+1} = \sum_j \frac{A_{ij}}{k_j^{out}} p_{j;n} \xrightarrow{\text{equilibrium}} p_i^* = \pi_i$$

$$\boxed{R(1) = \sum_{i,j} \left[\frac{A_{ij}}{k_j^{out}} \pi_j - \pi_i \pi_j \right] \delta(c_i, c_j) \neq Q}$$

$$\boxed{R(1) \neq Q(A) \quad \text{but} \quad R(1) = Q(Y)}$$

$$Y = \frac{X + X^T}{2} \qquad X_{ij} = \frac{A_{ij}}{k_j^{out}} \pi_j$$

Time as a resolution parameter

Let us consider a continuous-time random walk with Poisson waiting times

$$\dot{p}_i = \overbrace{\sum_j \frac{A_{ij}}{k_j} p_j - p_i}^{-L} \quad \xrightarrow{\text{equilibrium}} \quad p_i^* = k_i/2m$$

$$R(t) = \sum_{i,j}\left[\left(e^{t(B-I)}\right)_{ij}\frac{k_j}{2m} - \frac{k_i k_j}{(2m)^2}\right]\delta(c_i, c_j)$$

$$B_{ij} = A_{ij}/k_j$$

Probability that a walker is in the same community initially and at time t

Same probability for independent walkers

Time as a resolution parameter

Simpson diversity
Renyi entropy.

Let us consider a continuous-time random walk with Poisson waiting times

$$R(0) = 1 - \sum_{i,j} \frac{k_i k_j}{(2m)^2}\delta(c_i, c_j)$$

Communities = Single nodes

$$\boxed{1 - \sum_c \left(\frac{k_c}{2m}\right)^2} \quad k_c$$

$$R(t) \approx (1-t)R(0) + tQ_C \equiv Q(t)$$

Tuneable modularity of Reichart and Bornholdt

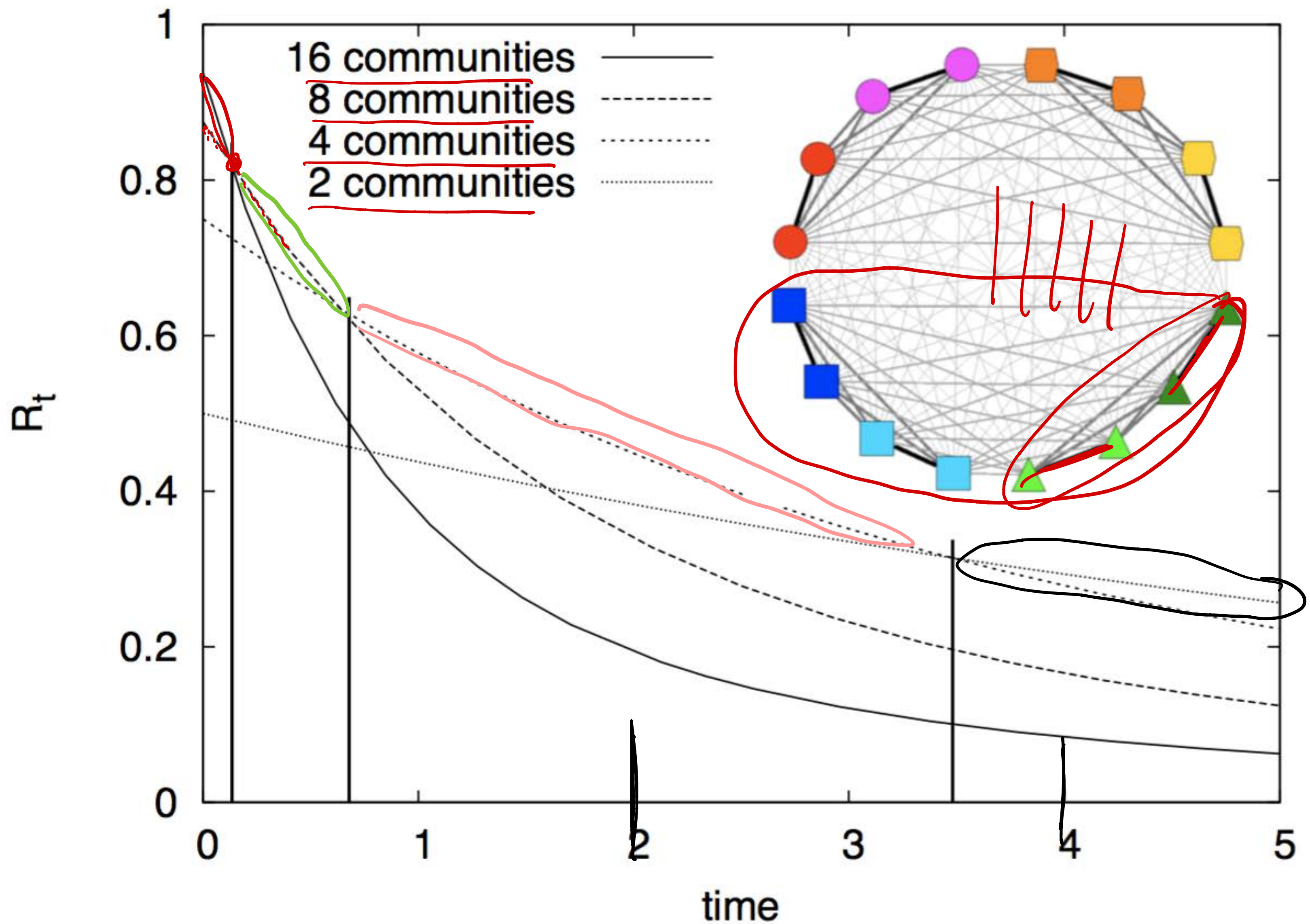
$$t \sim \frac{1}{\gamma} \qquad \sum_{ij}\left[A_{ij} - \gamma\frac{k_i k_j}{2m}\right]\delta(c_i, c_j)$$

time

Asymptotically, two-way partition given by the Fiedler vector

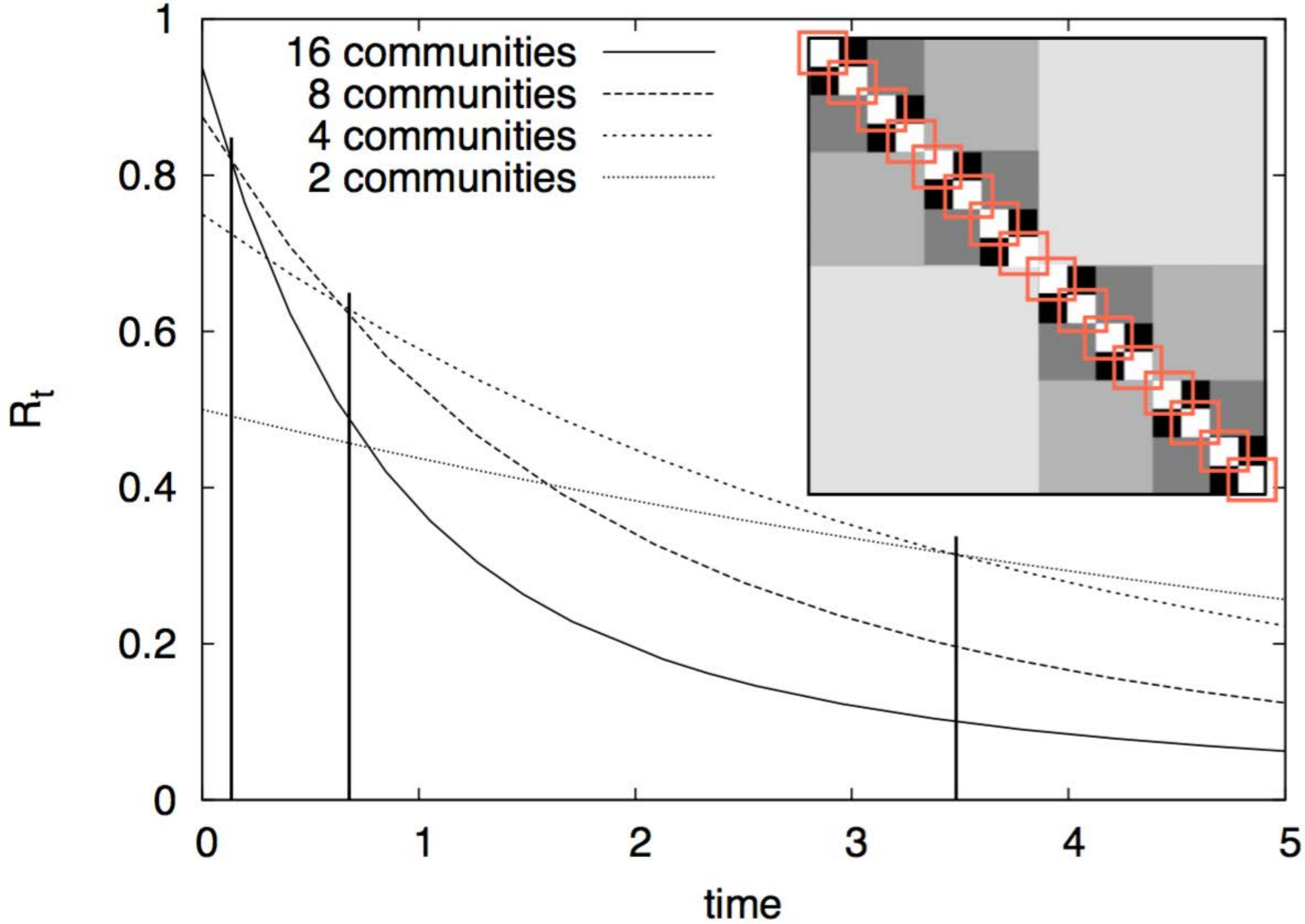
Time as a resolution parameter

Time is a “resolution parameter”: larger and larger communities when time is increased



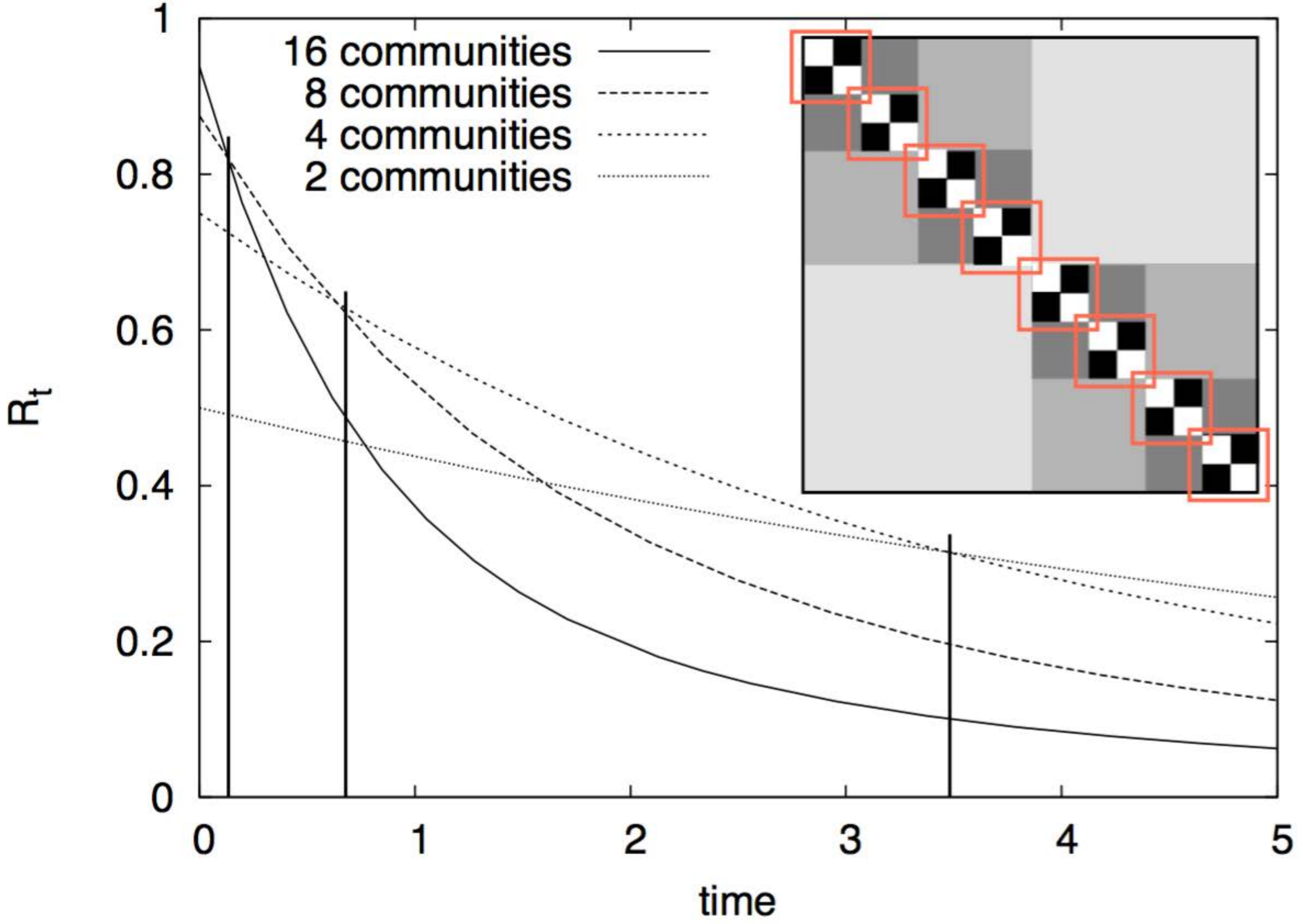
Time as a resolution parameter

Time is a “resolution parameter”: larger and larger communities when time is increased



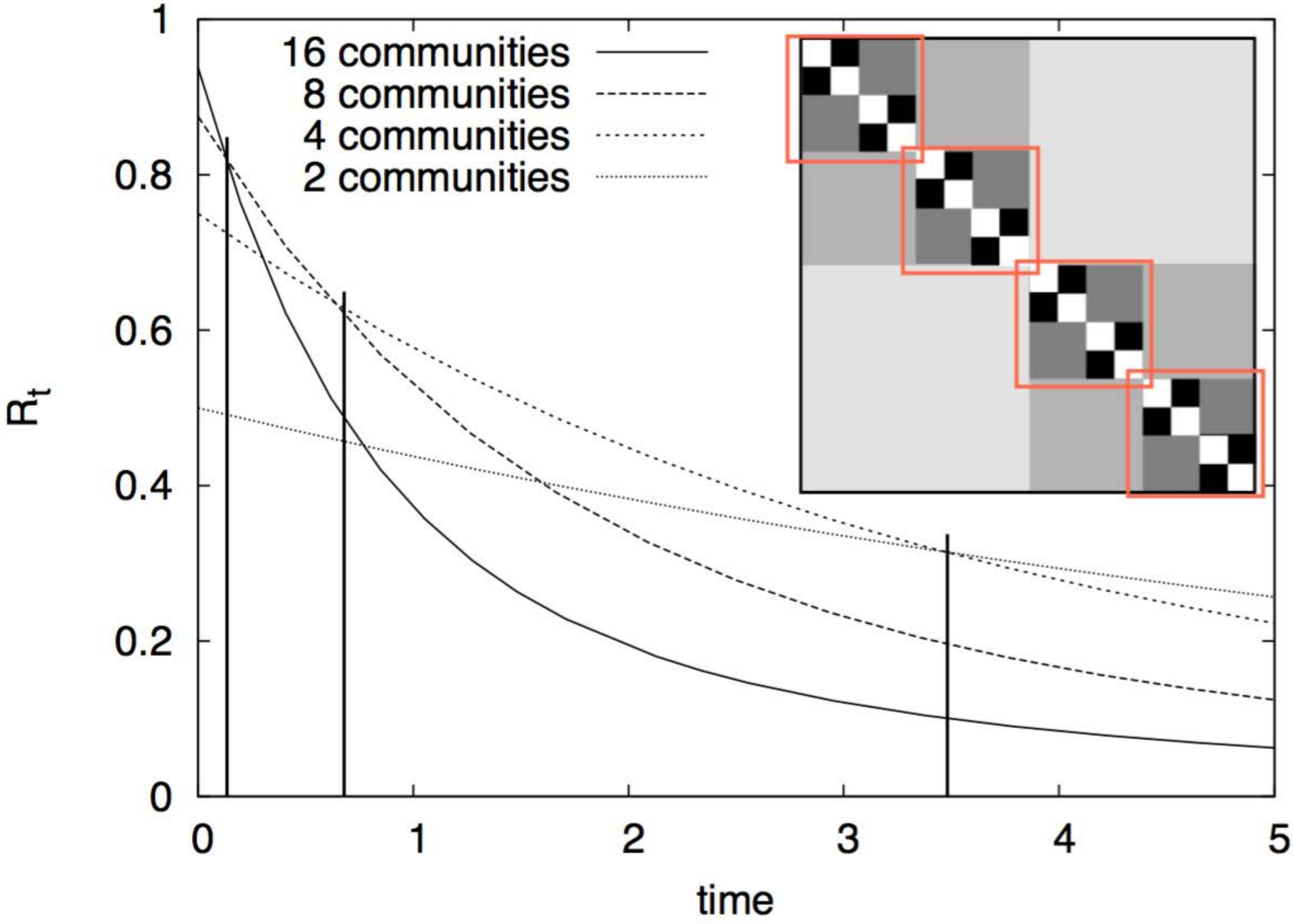
Time as a resolution parameter

Time is a “resolution parameter”: larger and larger communities when time is increased



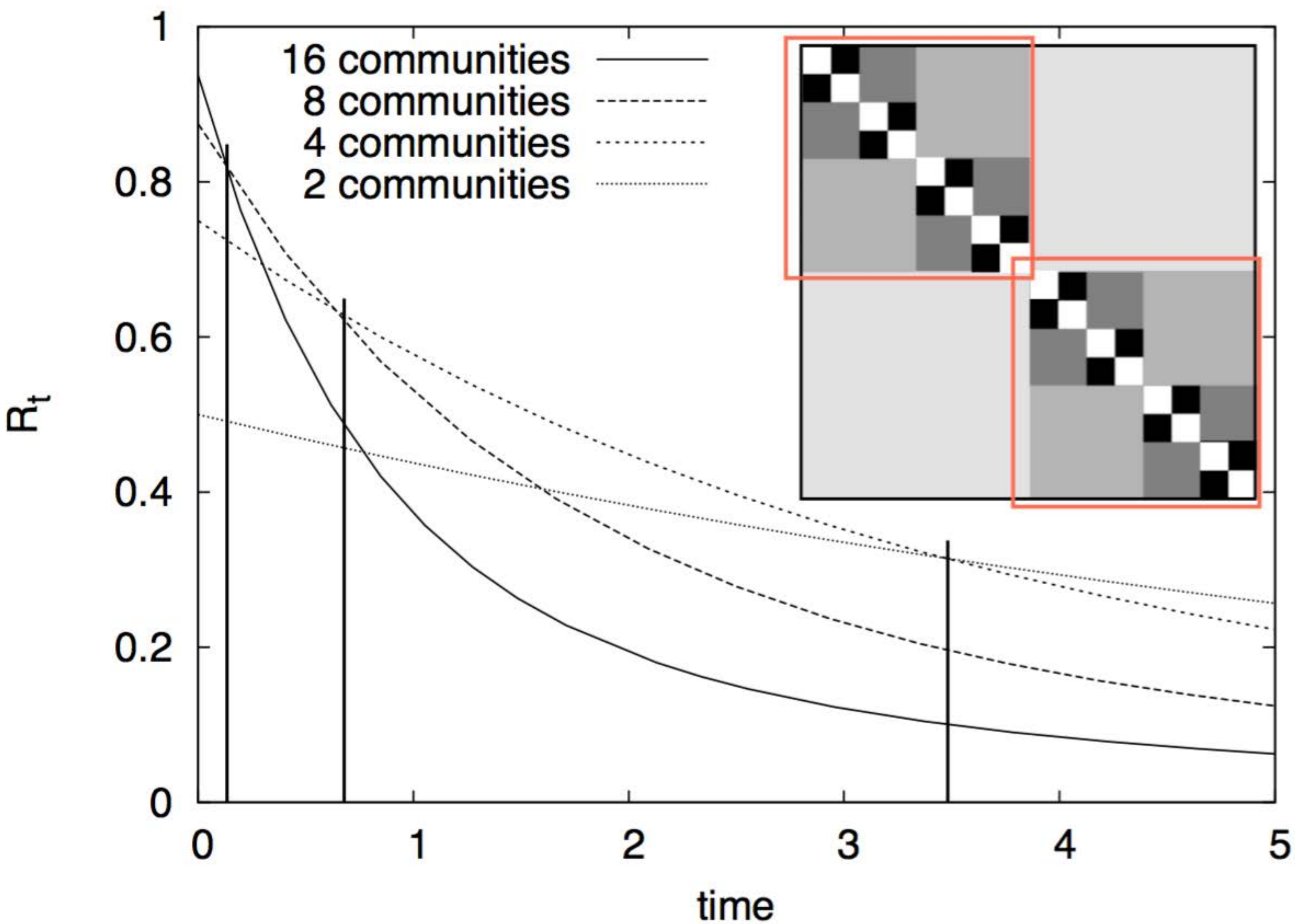
Time as a resolution parameter

Time is a “resolution parameter”: larger and larger communities when time is increased



Time as a resolution parameter

Time is a "resolution parameter": larger and larger communities when time is increased



In practice: optimization?

The stability R(t) of the partition of a graph with adjacency matrix A is equivalent to the modularity Q of a time-dependent graph with adjacency matrix X(t)

$$X_{ij}(t) = \left(e^{t(B-I)}\right)_{ij} k_j \qquad X_{ij}(t) = X_{ji}(t)$$

which is the flux of probability between 2 nodes at equilibrium and whose generalised degree is

$$\sum_j X_{ij}(t) = k_i$$

$$R(t) = \sum_{i,j} X_{ij}(t)/2m - k_i k_j/(2m)^2 \;\; \delta(c_i, c_j) = Q(X(t))$$

For very large networks: $\boxed{R(t) \approx (1-t)R(0) + tQ_C \equiv Q(t)}$

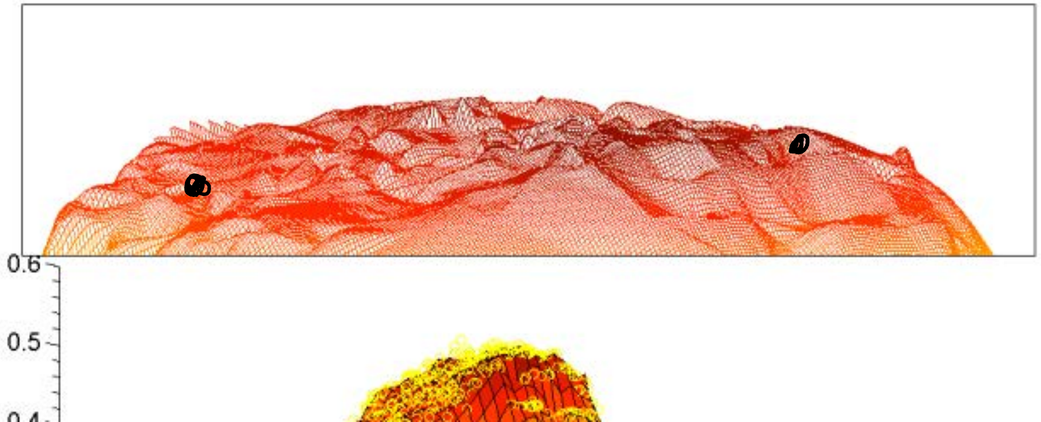
In practice: selection of the significant scales?

The optimization of R(t) over a period of time leads to a sequence of partitions that are optimal at different time scales.

How to select the most relevant scales of description?
The significance of a particular scale is usually associated to a certain notion of robustness of the optimal partition. Here, robustness indicates that a small modification of the optimization algorithm, of the network, or of the quality function does not alter this partition.

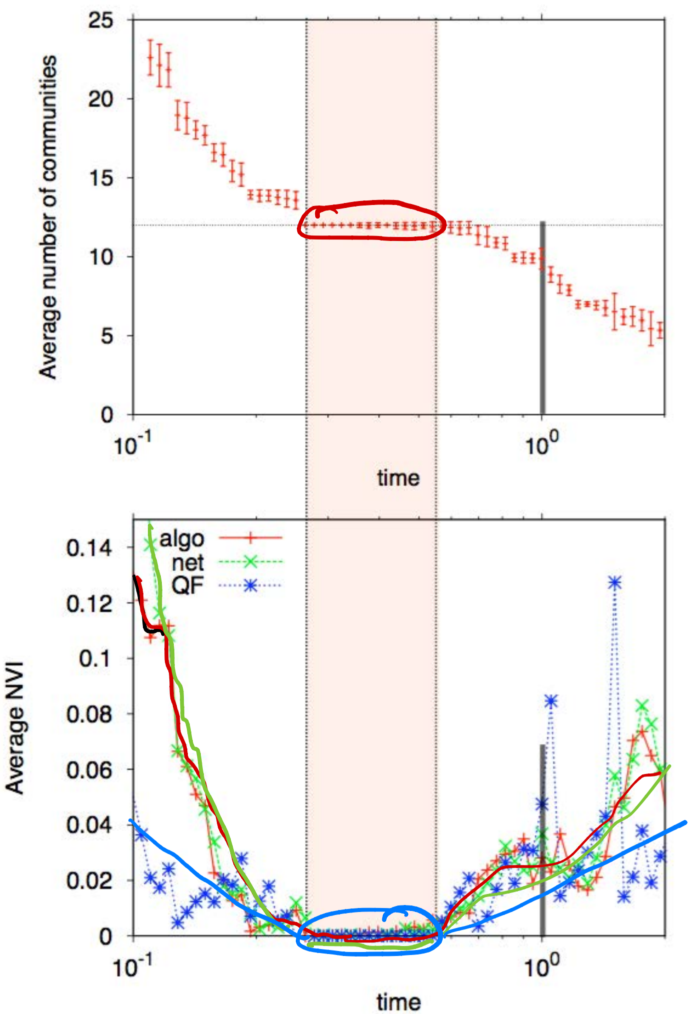
We look for regions of time where the optimal partitions are very similar. The similarity between two partition is measured by the normalised variation of information.

Intuition: at a bad scale, several competing maxima make the lanscape more rugged, leading to a sensitivity in the outcome of the algorithm



In practice: selection of the significant scales?

football



algo: for each t, 100 optimizations of Louvain algorithm while changing the ordering of the nodes

$$\langle V\rangle_{\text{algo}}(t) = \frac{2}{T(T-1)} \sum_{i=1}^{T} \sum_{i'=i+1}^{T} V(\mathcal{P}_i(t), \mathcal{P}_{i'}(t)).$$

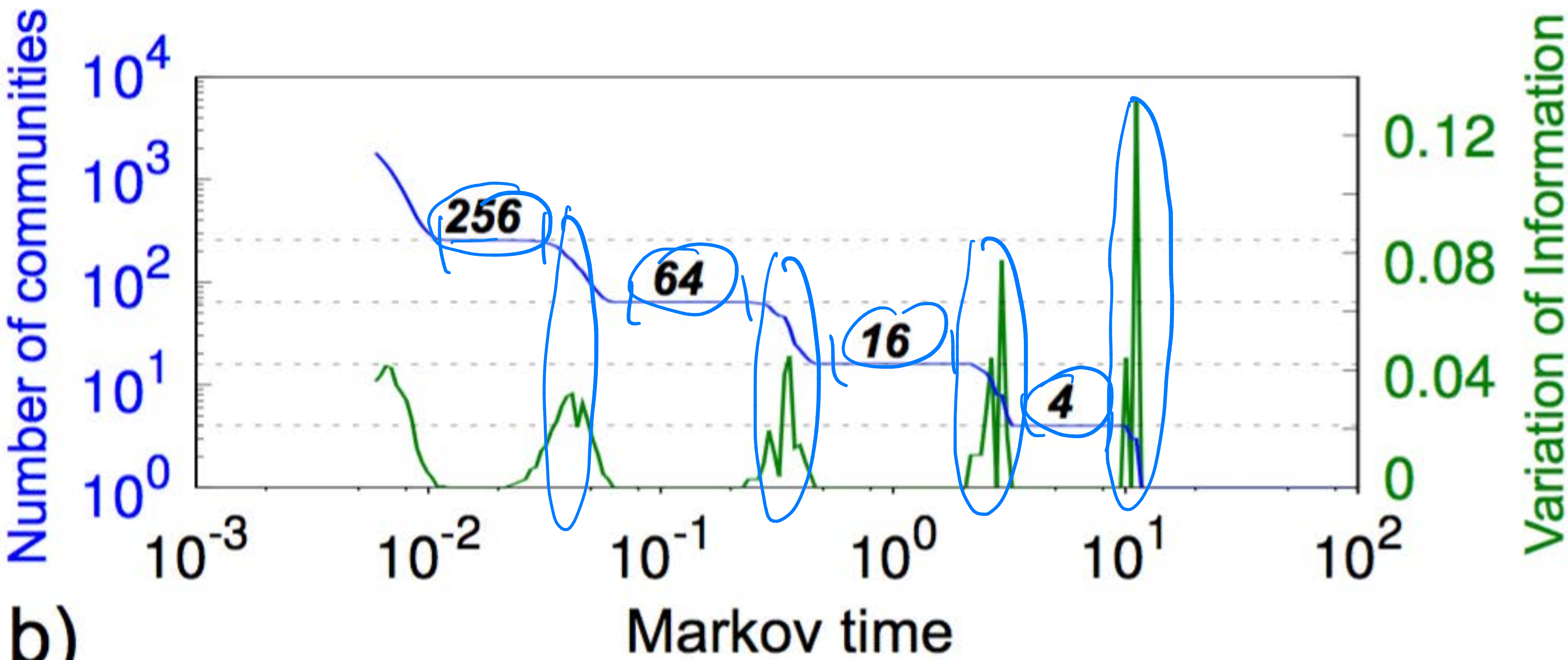
net: for each t, 100 optimizations with a fixed algorithm but randomized modifications of the network

QF: for each t, one optimization. Partitions at 5 successive values of t are compared.

Compatible notions of robustness:
Lack of robustness -> high degeneracy in the landscape: uncovered partitions are not to be trusted; wrong resolution

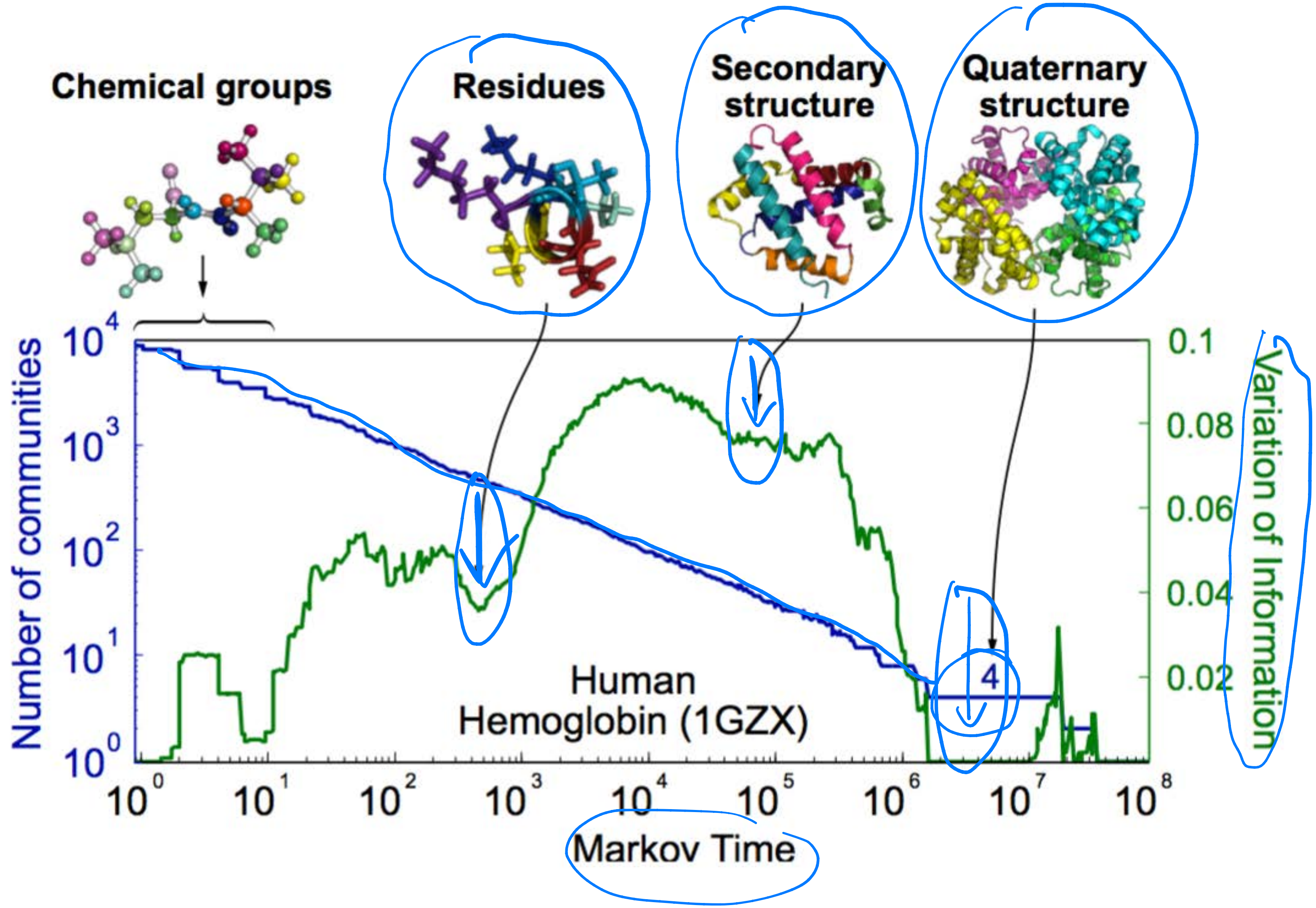
Time as resolution parameter

a)



b)

Time as resolution parameter



Time as resolution parameter

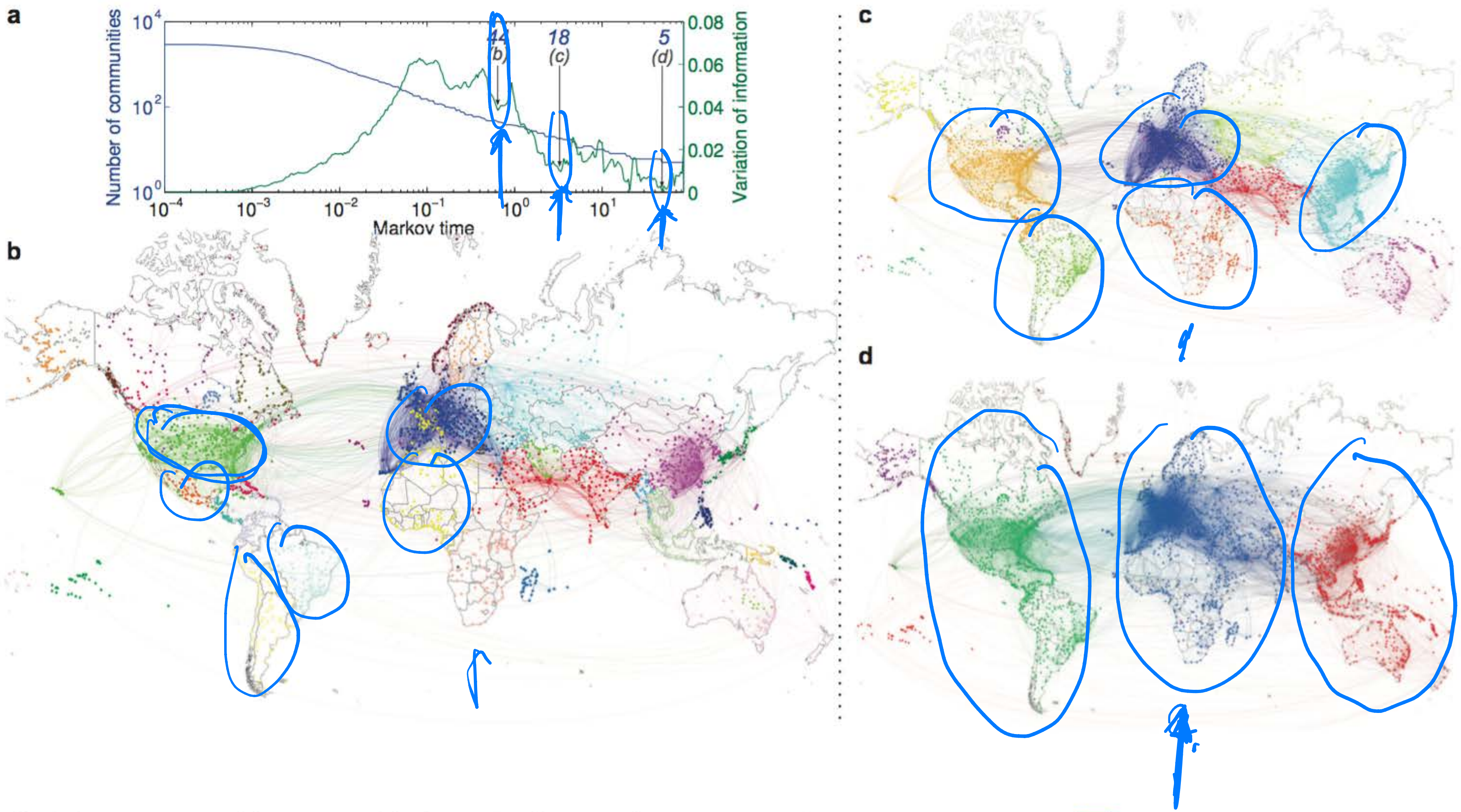
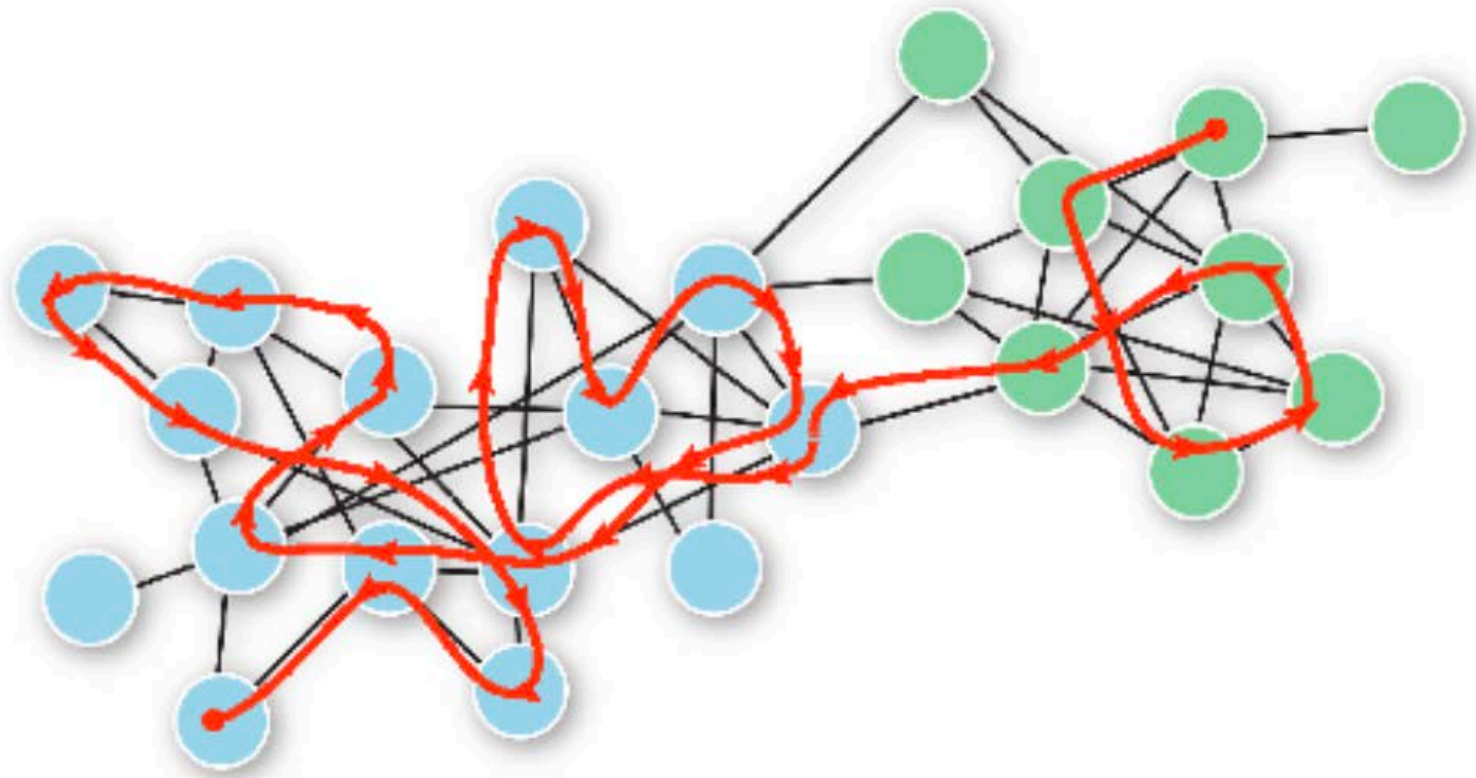


Fig. 8. **Flow communities at multiple scales in an airport network.** The airport network [82] contains $N = 2905$ nodes (airports) and 30442 weighted directed edges. The weights record the number of flights between airports (i.e., the network does not take into account passenger numbers, just the number of connections). Representative partitions at different levels of resolution with (b) 44, (c) 18 and (d) 5 communities are presented. The partitions correspond to dips in the normalized variation of information in (a) and show persistence across time (see Suppl. Info.).



(1) How does the modular structure of a network affect dynamics?
(2) How can dynamics help us characterise and uncover the modular structure of a network?

Some reading

Discussion on different principles behind community detection methods:
The many facets of community detection in complex networks, Michael T. Schaub, Jean-Charles Delvenne, Martin Rosvall, Renaud Lambiotte, Appl Netw Sci (2017) 2: 4

Excellent overview of the literature on community detection:
Doreian, P., Batagelj, V., & Ferligoj, A. (2020). Advances in Network Clustering and Blockmodeling. John Wiley & Hoboken, NJ.

The Map Equation:
Rosvall, M., & Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences, 105(4), 1118–1123.

This talk is based on:
Lambiotte, R., Schaub, M.
(2022) Modularity and Dynamics on Complex Networks, Cambridge University Press

Beyond assortative communities

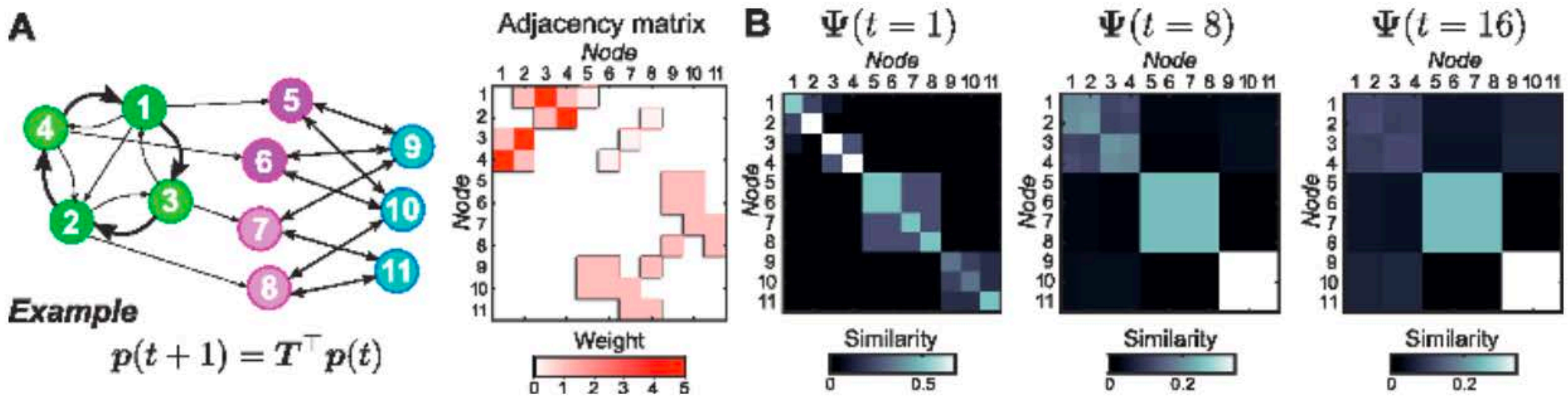


Figure 12 Dynamical similarity measures for random walks. A Visualisation of a (not strongly connected) directed network and its adjacency matrix. There are three main types of nodes identified by their different colours (subgroups within those three groups indicated by lighter colour). **B** The block structure in the similarity matrix $\boldsymbol{\Psi}(t) = \boldsymbol{T}^t[\boldsymbol{T}^t]^\top$, where $\mathcal{W}$ was chosen to be the identity matrix, identifies the dynamical role of the nodes at different times. Note that nodes within the cyan and violet groups are not connected to each other (i.e., the grouping is not assortative). Figure adapted and reproduced from Schaub et al. (2019a) with permission.

Connections to embedding techniques, such as Deepwalk, the notion of structural equivalence, of block models, etc.

Schaub, M. T., Delvenne, J.-C., Lambiotte, R., & Barahona, M. (2019a). Multiscale dynamical embeddings of complex networks. Physical Review E, 99(6), 062308. doi: https://doi.org/10.1103/PhysRevE.99.062308

Beyond assortative communities

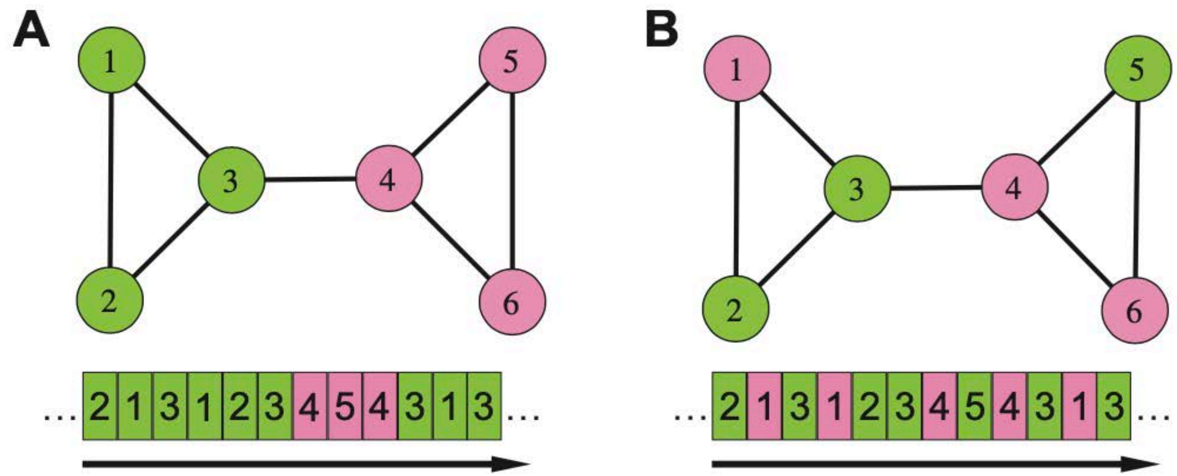


Figure 10 Markov stability and random walks. Given a partition of a graph, here illustrated by two colours, Markov stability is defined by the sequence of communities visited by the random-walk process. Intuitively, for a good partition as in A, the random walker will persist for long times inside a community before escaping it. Markov stability captures the persistence of a random walker at a timescale t via its clustered covariance matrix.

Instead at looking at the (linear) auto-correlation of the sequence of communities, you can consider auto-information.

Faccin, Mauro, Michael T. Schaub, and Jean-Charles Delvenne. "State aggregations in Markov chains and block models of networks." *Physical Review Letters* 127.7 (2021): 078301.

What about directed networks?

Dynamics is affected by the presence of hierarchies and communities

How directed is a directed network?

R. S. MacKay, S. Johnson, B. Sansom

A physical model for efficient ranking in networks

Caterina De Bacco, Daniel B. Larremore, Cristopher Moore

Urban spatial structures from human flow by Hodge-Kodaira decomposition

Takaaki Aoki, Shota Fujishima, Naoya Fujiwara

Even in the linear case, the spectral properties are more subtle.

Structure and dynamical behavior of non-normal networks

MALBOR ASLLANI, RENAUD LAMBIOTTE, AND, TIMOTEO CARLETTI Authors Info & Affiliations

Markov stability for temporal networks

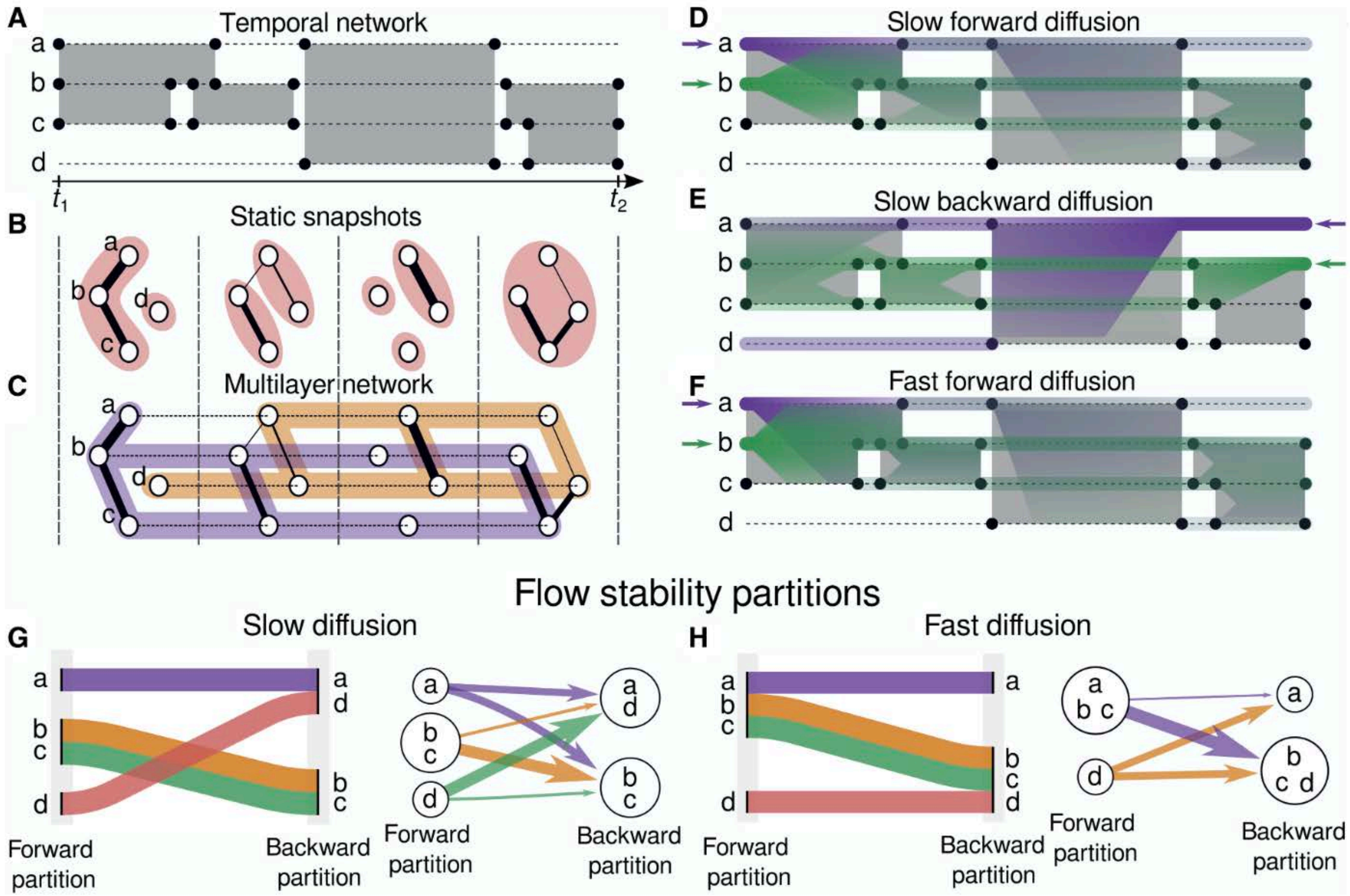
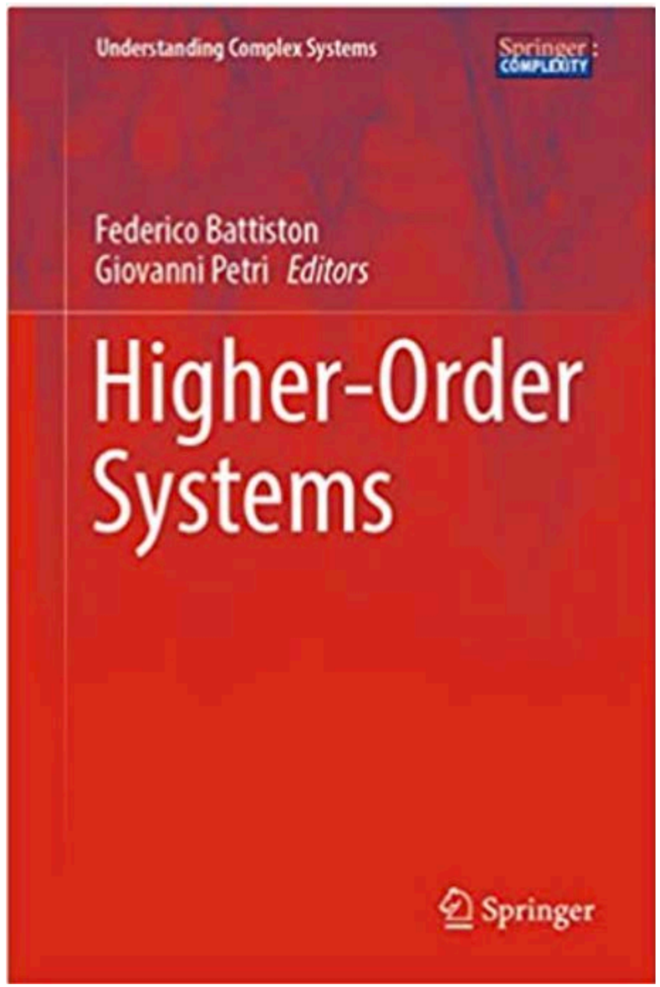


Fig. 1. Schematic representation of the flow stability compared to other temporal community detection methods.

Bovet, Alexandre, Jean-Charles Delvenne, and Renaud Lambiotte. "Flow stability for dynamic community detection." *Science Advances 2022*

Dynamics and community for “higher-order” networks

Random walk on hypergraphs, Hodge Laplacian, etc.



Lambiotte, Renaud, Martin Rosvall, and Ingo Scholtes. "From networks to optimal higher-order models of complex systems." Nature physics 15.4 (2019): 313-320.