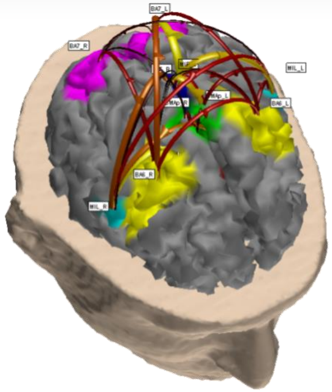
COMMUNITY DETECTION

Bioinformatics
ay 2018-2019

Manuela Petti

10/12/2018

Brain as a complex network

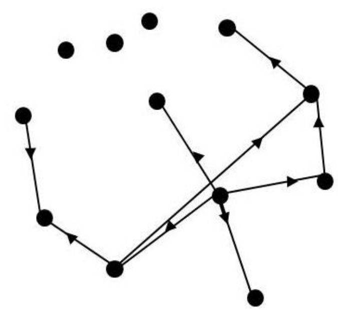


Brain connectivity

Information flows between different brain regions

Need to:

- **understand** the network organization
- **quantify** brain connectivity properties

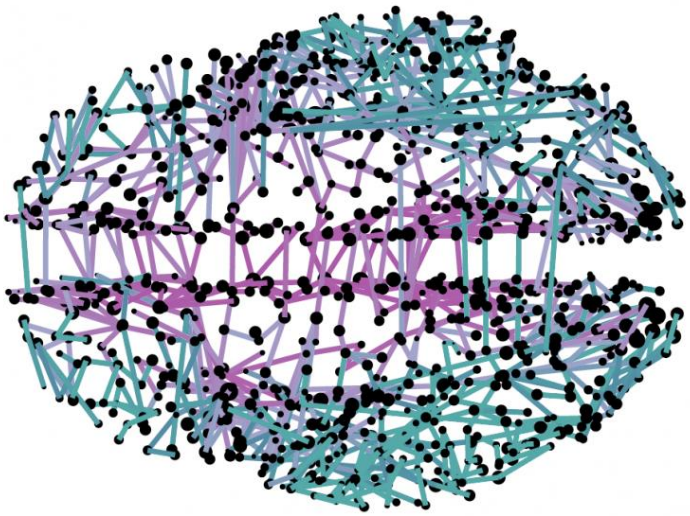


Graph theory

a **graph** is a mathematical object consisting of a set of **nodes** (brain areas) linked by means of **edges** (anatomical/functional connections)

Graph indices

Graph theory in Neuroscience



Microscopic

Degree (in-out)
Betweenness centrality
Closeness centrality

Mesoscopic

Motifs
Modularity

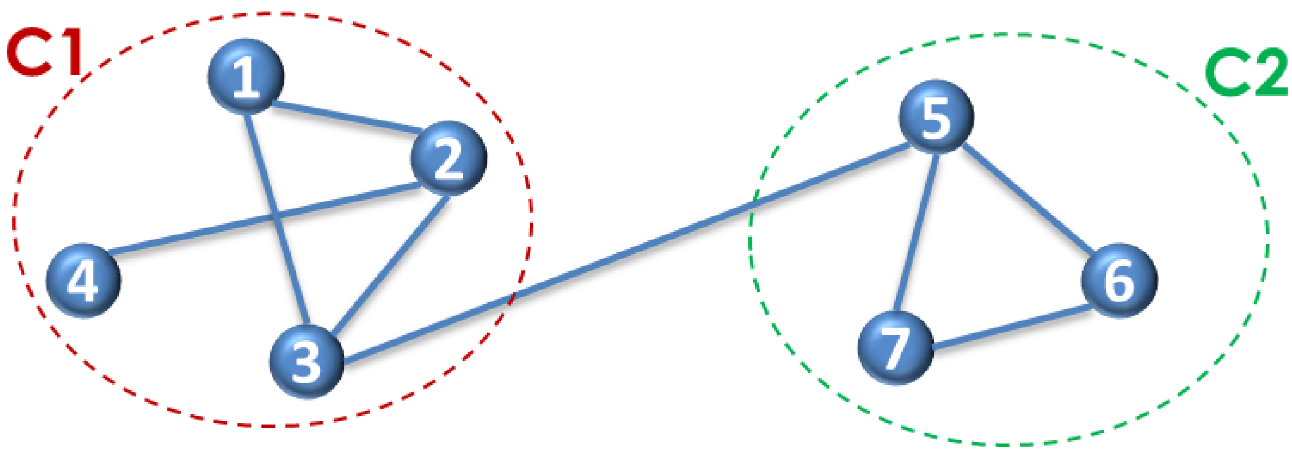
Macroscopic

Average Path length
Clustering coefficient
Global efficiency
Local efficiency

Community Detection in real Networks

Real networks properties:

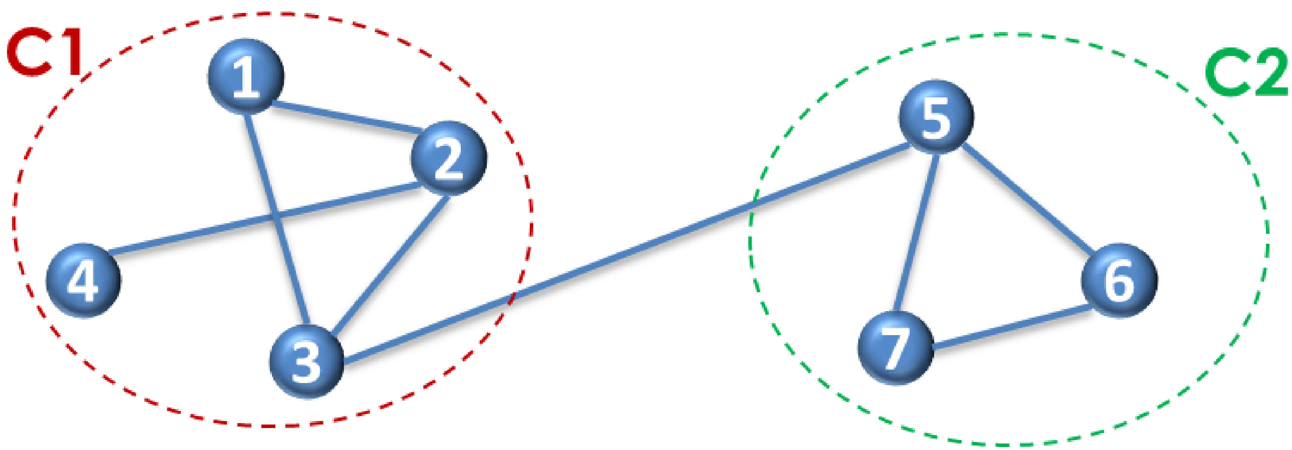
- Smallworldness
- Degree distribution
- Motifs
- <u>Community structure</u>



Community Definition

COMMUNITIES (OR COHESIVE GROUPS):

subsets of vertices within which vertex–vertex connections are dense, but between which connections are less dense



Community Detection in real Networks

In binary network, the identification of communities is possible only if **graphs are sparse**, i. e. if the number of edges *m* is of the order of the number of nodes *n*

If m>>n, the distribution of edges among the nodes is too homogeneous for communities to make sense

In weighted network, the identification of communities is possible with a **heterogeneous distribution of weights**.
→ communities are identified as subgraphs with a high internal density of weight

Quality function

Community detection based on a **quantitative criterion to assess the goodness of a graph partition**

A **quality function** is a function that assigns a number to each partition of a graph. In this way one can rank partitions based on their score given by the quality function.

high scores → “good” partitions
so the one with the largest score is by definition the best

The question of when a partition is better than another one is ill-posed, and the answer depends on the specific concept of community and/or quality function adopted

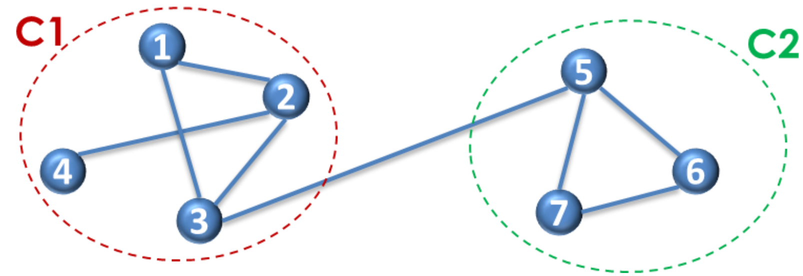
Community Detection in real Networks

No definition is
universally accepted

(Girvan and Newman, 2002):
Seminal paper in which Girvan and Newman proposed a new algorithm, aiming at the identification of communities and based on a centrality measure (edge betweenness)

Newman's Algorithm

Newman's algorithm is based on the quality function or "modularity" Q defined as the difference between intra-cluster links and the expected value of the same quantity if edges fall at random



$$Q = \sum_i (e_{ii} - a_i^2)$$

where:

- e_{ii} is the number of edges that fall within communities *i* normalized for the total number of connections in the network
- $a_i = \sum_j e_{ij}$
- e_{ij} is the fraction of edges in the network that connect vertices in group *i* to those in group *j*

M. E. J. Newman, *Physical Review E,2004*

Newman's Algorithm

$Q = 0$ → the division gives no more within-community edges than would be expected by random chance

high value of Q represents a good community division

⇩

Optimization of Q over all possible divisions to find the best one

VERY COSTLY!

M. E. J. Newman, *Physical Review E,2004*

Newman's Algorithm

Agglomerative hierarchical clustering method:

- each vertex is the sole member of community g (g=1,…,G)

$$G = N$$

- the algorithm repeatedly join communities together in pairs, choosing at each step the join that results in the greatest increase (or smallest decrease) in Q
- The progress of the algorithm can be represented as a "dendrogram"
- Cuts through this dendrogram at different levels give divisions of the network into larger or smaller numbers of communities and we can select the best cut by looking for the maximal value of Q

M. E. J. Newman, *Physical Review E,2004*

Newman's Algorithm

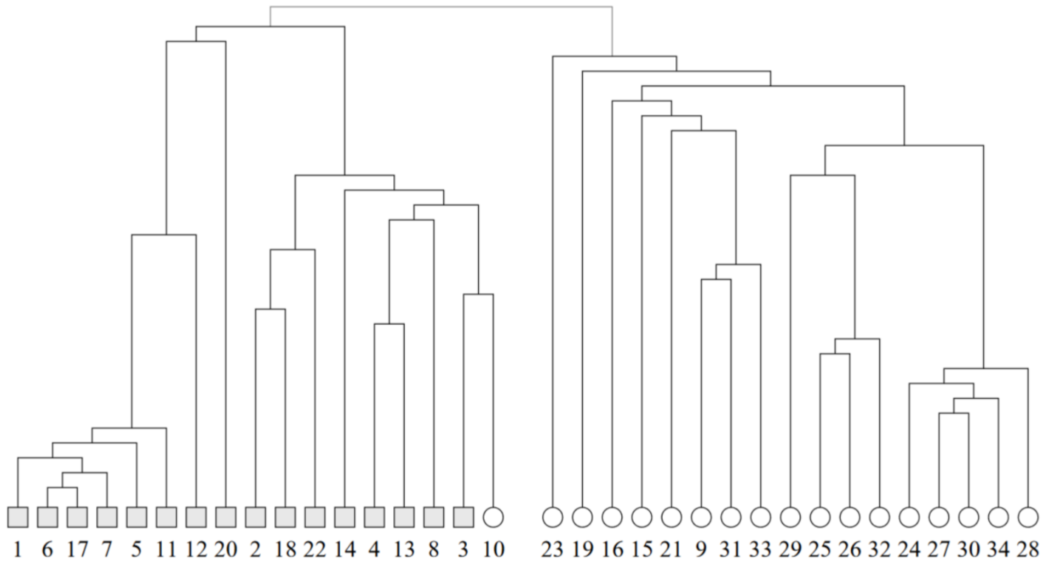


FIG. 2: Dendrogram of the communities found by our algorithm in the "karate club" network of Zachary [5, 17]. The shapes of the vertices represent the two groups into which the club split as the result of an internal dispute.

M. E. J. Newman, *Physical Review E,2004*

Leicht & Newman Algorithm

Method for the discovery of **communities in directed networks** that makes explicit use of the **information contained in edge directions**

Extension of the modularity optimization method for undirected networks (M. E. J. Newman, *Physical Review E, 2004)*

$$Q = \frac{1}{2m} \sum_{ij} (A_{ij} - P_{ij}) \delta(C_i, C_j)$$

where

- A is the adjacency matrix
- m is the total number of link in the network
- P_{ij} is the expected number of edjes between nodes i and j in the null model
- $\delta(C_i, C_j) = 1$ only if i and j belong to the same community ($C_i = C_j$)

Leicht & Newman, *PhysRevLett, 2008*

Leicht & Newman Algorithm

Consider two vertices A and B.

A → high out-degree
low in-degree

B → low out-degree (reverse situation)
high in-degree

↓

a given edge is more likely to run from A to B than viceversa because there are more ways it can fall in the first direction (A→B) than in the second (B→A)

If in the real network there is an edge from B to A, it should be considered a bigger surprise than an edge from A to B and hence should make a bigger contribution to the modularity, since modularity should be high for statistically surprising configurations

Leicht & Newman, *PhysRevLett, 2008*

Leicht & Newman Algorithm

An edge from vertex j to vertex I with probability:

$$P_{ij} = \frac{k_i^{in} k_j^{out}}{m}$$

where k_i^{in} and k_j^{out} are the in- and out-degrees of the vertices

Binary networks:

$$Q_d = \frac{1}{m} \sum_{ij} (A_{ij} - \frac{k_i^{out} k_j^{in}}{m}) \delta(C_i, C_j)$$

Weighted network

$$Q_{dw} = \frac{1}{W} \sum_{ij} (W_{ij} - \frac{s_i^{out} s_j^{in}}{W}) \delta(C_i, C_j)$$

Leicht & Newman, *PhysRevLett, 2008*

Limits of modularity

Modularity optimization has limits in resolution (Fortunato and Barthélemy, 2007): it may fail to identify modules smaller than a scale which depends on the total size of the network

Therefore not always the maximum value is indicative of a good partition

Louvain Algorithm

The algorithm finds high modularity partitions of **large networks** in **short time** and that unfolds a complete hierarchical community structure for the network, thereby giving access to **different resolutions of community detection**

The algorithm is divided in 2 phases that are repeated iteratively

Blondel et al, J. Stat. Mech, *2008*

Louvain Algorithm

1st phase

Given a weighted network of N nodes:

1) Assignation of a different community to each node of the network
2) for each node i, we consider the neighbours j and we evaluate the gain of modularity that would take place by removing I from its community and by placing it in the community of j
3) if this gain is positive, node i is placed in the community for which this gain is maximum

This process is applied repeatedly and sequentially for all nodes until no further improvement can be achieved and the first phase is then complete

This first phase stops when a local maxima of the modularity is attained, i.e., when no individual move can improve the modularity.

Blondel et al, J. Stat. Mech, *2008*

Louvain Algorithm

! the output of the algorithm can depend on the order in which the nodes are considered

The gain in modularity ΔQ is

$$\Delta Q = \left[\frac{\Sigma_{in} + k_{i,in}}{2m} - (\frac{\Sigma_{tot} + k_i}{2m})^2\right] - [\frac{\Sigma_{in}}{2m} - \left(\frac{\Sigma_{tot}}{2m}\right)^2 - (\frac{k_i}{2m})^2]$$

where

- Σ_{in} is the sum of the weights of the links inside cluster C
- Σ_{tot} is the sum of the weights of the links incident to nodes in C
- k_i is the sum of the weights of the links incident to node *i*
- $k_{i,in}$ is the sum of the weights of the links from *i* to nodes in C
- m is the sum of the weights of all the links in the network

Blondel et al, J. Stat. Mech, *2008*

Louvain Algorithm

2nd phase
Building a new network whose nodes are now the communities found during the first phase (super-nodes)

The weights of the links between the new nodes are given by the sum of the weight of the links between nodes in the corresponding two communities.
Links between nodes of the same community lead to self-loops for this community in the new network.

Once the second phase is completed, it is then possible to reapply the first phase of the algorithm to the resulting weighted network and to iterate.

The passes are iterated until there are no more changes and a maximum of modularity is attained

Blondel et al, J. Stat. Mech, *2008*

Louvain Algorithm: Advantages

- ✓ Easy to implement
- ✓ Extremely fast
- ✓ No resolution limit problem

Blondel et al, J. Stat. Mech, *2008*

Dynamic network

A dynamic network is defined as a series of network snapshots at two or more time points, where time can represent seconds, days, years or various states of a system

To perform community structure detection in dynamic network

Evolutionary clustering → the community structure of a network snapshot is identified by taking into account both its current state as well as previous time points

incorporating temporal information into network modelling frameworks may lead to more accurate representations of complex systems

Dynamic network

Some methods aim to detect drastic discontinuities in in community structure which represent some form of important ‘event’

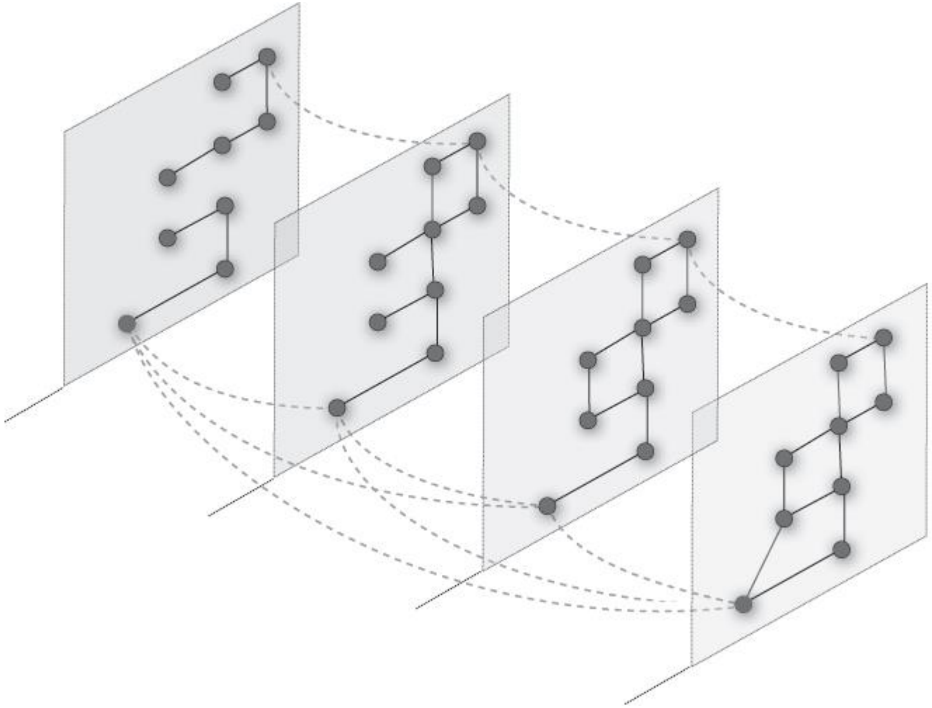
Other methods aim to extract the most stable and reliable community structure in a time interval

generalized Louvain Algorithm

Determination of community structure via quality functions to multislice networks that are defined by coupling multiple adjacency matrices

The connections encoded by the network slices are flexible:

- variations across time
- variations across different types of connections
- community detection of the same network at different scales



Mucha et al, Science, *2010*

generalized Louvain Algorithm

Robust algorithm with a modified version of modularity designed for multilayer networks

Multi-layer (multi-slice) modularity function

$$Q = \frac{1}{2\mu} \sum_{ijlr} \{(A_{ijl} - \gamma_l P_{ijl})\delta_{lr} + \delta_{ij}\omega_{jlr}\} \delta(g_{il}, g_{jr})$$

Interlayer coupling parameter (temporal resolution parameter)

Mucha et al, Science, *2010*

generalized Louvain Algorithm

$$Q = \frac{1}{2\mu}\sum_{ijlr}\{(A_{ijl} - \gamma_l P_{ijl})\delta_{lr} + \delta_{ij}\omega_{jlr}\}\delta(g_{il}, g_{jr})$$

A_{ijl}: elements of adjacency matrix at layer *l*
P_{ijl}: components of the corresponding layer-*l* matrix for the optimization null mode *l*
γ_l: structural resolution parameter of layer *l*
g_{il}: community assignment of node *i in layer l*
g_{jr}: community assignment of node *j in layer r*
ω_{jlr}: interlayer coupling parameter from node *j* in layer *r* to node *j* in layer *l*
$\mu = \frac{1}{2}\sum_{jr}\kappa_{jr}$*: t*otal edge weight in the network with:

$\kappa_{jl} = k_{jl} + c_{jl}$: strength of node *j* in layer *l*
$k_{jl} = \sum_i A_{ijl}$: intra-layer strength of node *j* in layer *l*
$c_{jl} = \sum_r \omega_{jlr}$: inter-layer strength of node *j* in layer *l*

Mucha et al, Science, *2010*

The Map Equation

The map equation method proposed by Rosvall and Bergstrom in 2008, known as **Infomap**, identifies communities according to information flow in the networks

Since communities can be thought composed by interdependent interactions where interaction between components A and B influences the interaction between components B and C, and so on, we can suppose that there is a flow of some entity that connects the system components and generates these interdependencies

Local interactions induce a **system-wide flow of information** that characterizes the behavior of the full system

to understand the flow of information on the network

The Map Equation

Huffman codes to gives short codewords for commonly visited nodes, and long codewords for rarely visited nodes.

$$L(M) = q_{\curvearrowright} H(\mathcal{Q}) + \sum_{i=1}^{m} p_{\circlearrowright}^{i} H(\mathcal{P}^{i})$$

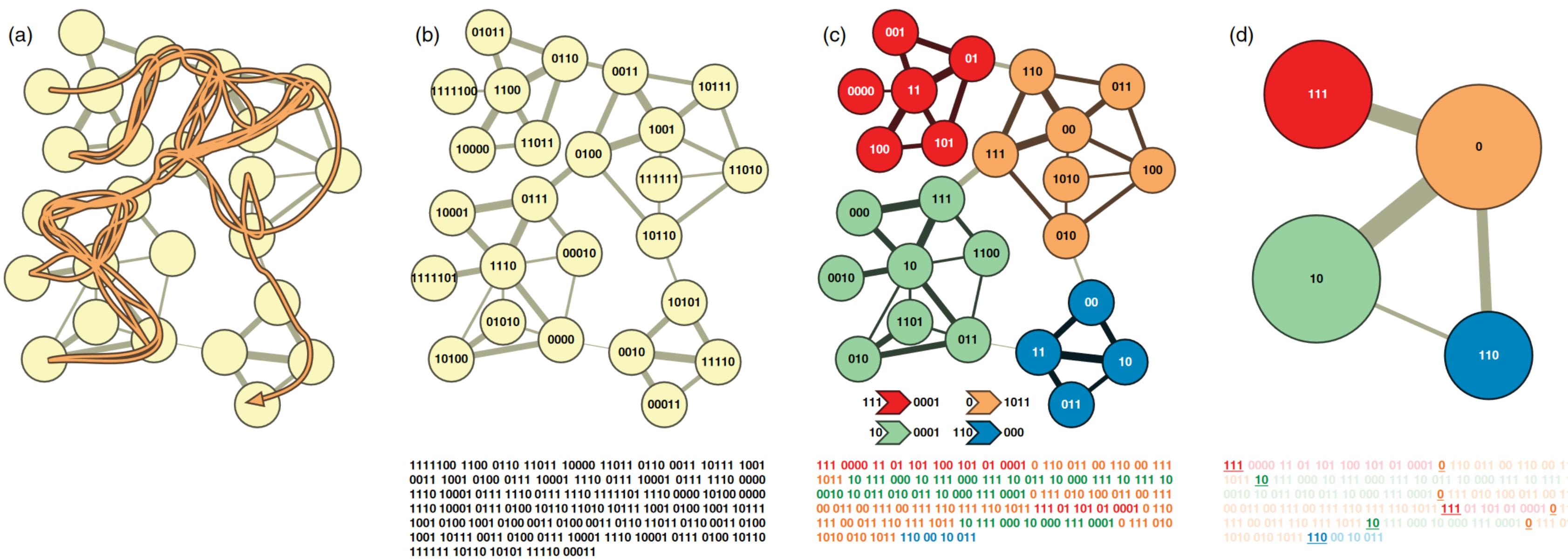
This equation has two parts:

- the first one is to explain the movements between the communities, where *q* is the probability that a random walker switches communities and *H(C)* is the entropy of the community index codewords.
- the second part explains movements within the communities, where *pi* is the fraction of the movements within community *ci* and $H(P^i)$ is the entropy of the movements within community *ci*

Detecting communities by compressing the description of information flows on networks

Rosvall and Bergstrom, PNAS, *2008*

The Map Equation



Rosvall and Bergstrom , PNAS, *2008*

The Map Equation

Infomap is based on the **principles of information theory**

Problem of finding the optimal clustering of a graph as the problem of finding a description of minimum information of a **random walk** on the graph

The algorithm maximizes an objective function called the **Minimum Description Length**

An acceptable approximation to the optimal solution can be found **quickly**

Previous studies have found Infomap's performance to remain stable for **networks with up to 100,000 nodes**

The Map Equation

http://www.mapequation.org/code.html