European Society of Computational Methods in Sciences and Engineering (ESCMSE)

Journal of Numerical Analysis, Industrial and Applied Mathematics (JNAIAM)
vol. 6, no. 1-2, 2011, pp. 29-50
ISSN 1790–8140

Random Walks on Complex Modular Networks[1][2]

Natasa Djurdjevac[3], Sharon Bruckner, Tim OF Conrad, Christof Schütte

Fachbereich Mathematik und Informatik
Institut für Mathematik
Freie Universität Berlin
{*djurdjev,sharonb,conrad,schuette*}@math.fu-berlin.de

Received 20 February, 2011; accepted in revised form 10 March, 2011

Abstract: Complex modular networks appear frequently, notably in the biological or social sciences. We focus on two current challenges regarding network modularity: the ability to identify (i) the modules of a given network, and (ii) the hub states as nodes with highest importance in terms of the communication between modules. Our approach towards these goals uses random walks as a mean to global analysis of the topology and communication structure of the network. We show how to adapt recent research regarding coarse graining of random walks. The resulting algorithms are based on spectral analysis of random walks and allow (A) an optimal identification of fuzzy assignments of nodes to modules, (B) computation of the fraction of the overall communication between modules supported by certain nodes, and (C) determination of the hubs as the nodes with the highest communication load.
Dedicated to Peter Deuflhard on the Occasion of his 65th Birthday.

© 2011 European Society of Computational Methods in Sciences and Engineering

Keywords: random walks, complex networks, modules, hubs, metastability, transition path theory, clustering

Mathematics Subject Classification: 05C81 Random walks on graphs

PACS: 05.40.Fb

1 Introduction

With the increasing power of high throughput technologies and storage capacities, more and more datasets from real-world systems in the form of complex networks become available. Similarly, the area of network analysis has expanded rapidly and attracted a lot of attention over the last 10 years. Networks are now widely recognized not only as outcomes of complex interactions, but as key determinants of structure, function, and dynamics in systems that span the biological, physical, and social sciences [1, 37, 38].

The so called "new science of networks" [4] has introduced novel paradigms of basic system properties, such as scale-free networks [5], small-world structure [50], and the importance of motifs [35] and organization into modules [21].

[1]Supported by the DFG research center Matheon "Mathematics for key technologies" in Berlin.
[2]Published electronically May 15, 2011
[3]Corresponding author. Supported by the Graduate School "Berlin Mathematical School".

Complexity Reduction by Finding Modules. Describing systems as abstract networks is a powerful tool [38], but even as abstractions they remain highly complex. Approaches to reducing the complexity for networks include characterizing the network in terms of simple statistics such as degree distributions [24], clustering coefficients [50], subgraphs overrepresented relative to an assumed null model (motifs) [3], and modules [39]. In the case of modules, networks are coarse-grained into clusters of nodes where nodes belonging to one cluster are highly interconnected, but have relatively few connections to nodes in other clusters.

Two current challenges regarding network modularity are (i) the ability to quantify to what extent a given network is "modular" [36, 39] and (ii) the ability to identify the modules of a given network. In his review article, Newman [37] summarizes a range of techniques for identifying modules. Some of these ideas have been applied to biological networks, and the results thus far are mixed but promising.

Biological Networks. The analysis of biological networks is still a very challenging but fascinating task, due to the complex, non-random organization and the diverse dynamic behaviours of the underlying systems. The topology of a large number of biological networks has been shown to be based on a scale-free degree distribution, such as protein-protein interaction networks or metabolic networks. This implies the existence of highly connected network hubs [5, 6], which may serve as central distributing elements or linkage points for many regions of a network [6, 23]. For example, in an early study by Fell and Wagner the authors found the metabolites with the highest degree of connectivity to be the core of *E. coli* metabolism [18]. Another study by Jeong et al. found that the ranking of the most connected metabolites is largely identical for all organisms [24].

Since biological systems are often organized in network modules [19, 39, 42, 20] hubs are also very often the structural key elements connecting these modules. These modules often represent a specific function, e.g. a specific synthesis pathway in a metabolic reaction network [43].

In this paper we are mainly interested in two things:
Identifying hubs: In general, nodes are called *hubs* if the have a high degree or a high centrality.[4] We will see later that these concepts are not sufficient and will give a more precise definition. However, in the vast majority of biological networks that exhibit a scale-free structure, hub-nodes dominate the topology and are usually of great biological significance.
Identifying modules: Modules are characterized by a higher frequency of connections within than between modules. Finding these module can help to decompose the complex network structure into functional sub-units that can be analysed in more detail in subsequent stages [19, 41, 45].

Random Walks. During the last years, new strategies have been developed for studying complex networks. Particularly, the method of random walks, as a fundamental dynamic process [22] has been well-established for structural analyses of networks, as it can fully account for local as well as global topological structure within the network [49, 40] and it is very useful for finding central nodes which can be used to identify actual hubs [2, 28, 40, 44]. The random walker defines a Markov chain on the state space that is given by the network's nodes, for details see below, Section 2. Therefore, partitioning a network into modules is tantamount with partitioning of losely connected aggregates of almost uncoupled Markov chains. Since there is a rich literature addressing different variants of the latter problem [32, 25, 7, 13, 31, 30, 8, 34, 29, 10] we also have a rich collection of possible approaches for addressing the former problem. The survey [26] will be our starting point; it outlines some of the fundamental connections between the structure of the network and kinetic properties of the random walker. However, the discussion in [26] and most of the aforementioned articles

[4] We identify centrality with the number of shortest paths between modules.

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)

does *not* address the following problem: A full partition of the network into modules may not be appropriate, since the *interfaces* between modules play an important role these interfaces will typically contain the nodes that are central to communication between nodes, i.e., the hubs. In a full partition, however, interface states will always be assigned to exactly one module instead of being identified as intermediate states between modules. In this contribution to this discussion we will show how to generalize the random walker approach to modular networks by allowing a *partial* partition of the network into modules and additional identification of interface states. Such random walker based partial or fuzzy partitions have recently been discussed in the literature [10, 47, 27] but did *not* address the identification of hub states in the interface. We will address especially this problem.

Outline. We first introduce and review the theoretical background of the random walker based approach to modular networks in Section 2. Next we discuss full and fuzzy partition based on that approach and relate it to the identification of modules in Section 3. Section 4 describes our new approach on how to identify interface states between modules and hubs as the most important interface states. Our theoretical investigations are complemented by numerical illustrations in Section 5.

2 Networks and Random Walkers

Let $G(V, E, w)$ be a network (or a finite weighted graph) with $|V| = n$ nodes, where $E \subset V \times V$ denotes the edge set, and the non-negative weights $w(x, y)$ satisfy $w(x, y) = 0$ if $(x, y) \notin E$. The most simple example of the weight matrix is $w(x, y) = 1$ for all $(x, y) \in E$.

In the following we will assume that the network under consideration is symmetric in the sense that $w(x, y) = w(y, x)$, i.e., the network is essentially undirected. We moreover assume that the network is connected, i.e., every node can be reached from every other node.

One can relate the network to a discrete-time Markov chain (X_k) with stochastic matrix P with entries $p(x, y)$ given by

$$p(x, y) = \frac{w(x, y)}{d(x)}, \quad d(x) = \sum_{y \in V} w(x, y).$$

The random walker defined by the Markov chain then moves from node to node randomly according to the transition probabilities

$$\mathbb{P}[X_1 = y | X_0 = x] = p(x, y).$$

Since the network is connected the random walker has a unique positive invariant measure μ, that is, if the walker starts μ-distributed then it is again μ-distributed after one step, $\sum_{x \in V} \mu(x)p(x, y) = \mu(y)$. Its explicit form

$$\mu(x) = \frac{d(x)}{\sum_{y \in V} d(y)},$$

allows us to observe immediately that the symmetry of the network implies that the walker is *reversible* in time, or, in other words, that the detailed balance condition $\mu(x)p(x, y) = \mu(y)p(y, x)$ is satisfied. As a consequence, its transition matrix is equivalent to a symmetric matrix. More precisely, by defining the weighted scalar product on

$$\langle u, v \rangle_\mu = \sum_{x \in V} u(x)v(x)\mu(x),$$

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)

we find $\langle u, Pv\rangle_\mu = \langle Pu, v\rangle_\mu$. Therefore, the spectrum of P is real-valued and thus can be ordered as follows:

$$1 = \lambda_1 > \lambda_2 \geq \lambda_3 \geq \dots$$

where the λ_j are the eigenvalues, $Pu_j = \lambda_j u_j$, associated to the eigenvectors u_j that have to be orthogonal, $\langle u_j, u_k\rangle_\mu = \delta_{jk}$. Thus we can write the kth-power of P in the form

$$P^k = \sum_{j=1}^{n} \lambda_j^k \langle u_j, \cdot\rangle_\mu \, u_j. \tag{1}$$

Since P^k describes the k-step transition probabilities of the walker,

$$\mathbb{P}[X_k = y | X_0 = x] = P^k(x, y),$$

equation (1) means that the eigenvalues λ_j imply the timescales $T_j = 1/|\log \lambda_j|$ of all relaxation processes of the random walker on the network, starting with the trivial timescale $T_1 = \infty$ on which the random walker relaxes to its invariant measure, via the slowest non-trivial scale T_2 to shorter and shorter relaxation timescales. If some eigenvalues, say $\lambda_1, \dots, \lambda_m$, are particularly close to 1 (significantly closer to 1 in modulus than all others), then the associated timescales are very long and significantly longer than all other relaxation timescales. These eigenvalues will be called *leading* or *dominant* in the following. Then the subspace spanned by the associated eigenvalues,

$$U_m = \mathrm{span}\{u_1, \dots, u_m\},$$

allows a low-dimensional approximation of the long-term behavior of the walker on the network:

$$P^k \approx \sum_{j=1}^{m} \lambda_j \langle u_j, \cdot\rangle_\mu \, u_j = (\Pi P \Pi)^k,$$

where Π denotes the orthogonal projection of P onto U_m. Therefore, one often considers the so-called *diffusion map* $\Phi_m : V \to \mathbb{R}^{m-1}$,

$$\Phi_m(x) = \Big(\lambda_2 u_2, \dots, \lambda_m u_m\Big),$$

that embeds the network into $\mathbb{R}^{m-1}$ [25].

3 Modules and Metastable Sets

Modules are parts of the network in which the nodes are more densely connected to each other than to other parts of the network. What does this mean in terms of properties of the random walker? In order to answer this question, let us introduce the natural transition probabilities between sets,

$$p(A, B) = \mathbb{P}_\mu(X_1 \in B | x_0 \in A) = \sum_{x \in A, y \in B} \mu(x) p(x, y),$$

that is, the probability that the walker, after having started in set $A \subset V$ distributed according to the invariant measure, will be found in set $B \subset V$ after one step. A module, being defined by the property of being connected internally more densely than externally, can thus be described as a subset $M \subset V$ of the nodes for which the transition probability $p(M, M^c)$ of the random walker from the module to its complement $M^c = V \setminus M$ is significantly small, or, vice versa, the residence probability $p(M, M) = 1 - p(M, M^c)$ significantly close to 1. In the Markov chain theory such sets are called *metastable sets* [9, 10]. However, our definition is lacking the necessary precision, since "significantly small" will not be sufficient for the identification of the modules of a network. In order to change this, we will discuss two different concepts for identifying modules.

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)

3.1 Complete modular partition

Let us first consider a *full partition* of the network into modules, meaning, let us consider m disjoint sets $M_1, \ldots, M_m$ that partition V, $\cup_{j=1}^{m} M_j = V$. The best possible partition into modules thus has to *maximize* the joint metastability of the sets,

$$\mathcal{D}(M_1, \ldots, M_m) = \sum_{j=1}^{m} p(M_j, M_j).$$

Markov chain theory [48, 46] provides us with the following lower and upper bound for the functional $\mathcal{D}$ for arbitrary partitions $M_1, \ldots, M_m$:

$$(1-\delta_m)^2 \cdot \lambda_m + \ldots + (1-\delta_2)^2 \cdot \lambda_2 + \lambda_1 + c \leq \mathcal{D}(M_1, \ldots, M_m) \leq \lambda_m + \ldots + \lambda_1, \tag{2}$$

where $c = -\lambda_{m+1}(\delta_m^2 + \ldots + \delta_2^2)$, and δ_j is the error of the projection of the eigenvector u_j onto the space spanned by the characteristic functions $\mathbf{1}_{M_j}$ of the sets,

$$\delta_j = \|(\mathrm{Id} - Q)u_j\|_{2,\mu}, \quad Q \text{ projection onto } D = \mathrm{span}\{\mathbf{1}_{M_1}, \ldots, \mathbf{1}_{M_m}\}, \tag{3}$$

where the projection is orthogonal with regards to the μ-weighted scalar product, and $\|\cdot\|_{2,\mu}$ denotes the associated norm. According to this result, we will find the optimal or at least an almost optimal partition by minimizing the projection errors δ_j. We will return to this important aspect later. For an algorithmic realization of this approach please visit [9, 10].

But first let us explore an equivalent formulation of the above maximization problem for finding the optimal sets in terms of the diffusion map Φ. To this end, let us follow [26] and consider the so-called dimension-j *centroid* $\Phi_j(M)$ of sets $M \subset V$ in terms of the M-average of the first j eigenvectors u_i, $i = 1, \ldots, j$,

$$\bar{u}_i(M) \quad = \quad \frac{1}{\mu(M)} \sum_{y \in M} \mu(y) u_i(y), \quad \Phi_j(M) = (\lambda_1 \bar{u}_1(M), \ldots, \lambda_j \bar{u}_j(M)),$$

with $\mu(M) = \sum_{x \in M} \mu(x)$ and the so-called *average diffusion distance* from the centroid within M

$$d^2(M) = \sum_{x \in M} \mu(x) \|\Phi_n(x) - \Phi_n(M)\|^2,$$

where $\|\cdot\|$ denotes the usual Euclidean 2-norm in $\mathbb{R}^n$, and $\Phi_n(x)$ is the diffusion map introduced above but in dimension n. Then, the so-called *diffusion map minimization*

$$\min_{[M_1, \ldots, M_m]} \sum_{i=1}^{m} d^2(M_i), \tag{4}$$

is equivalent [26] to

$$\max_{[M_1, \ldots, M_m]} \sum_{j=1}^{m} p(M_j, M_j).$$

This result shows that instead of trying to find the optimal metastable sets (the last maximization problem) we can project the network into the Euclidean space $\mathbb{R}^n$ by means of the diffusion map, and then solve the *clustering problem* (4). An algorithm for solving (4), a specific variant of the well-known k-means clustering algorithm, has been devised in [26].

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)

A further way to identify optimal partition sets by means of a minimization principle can be derived by considering the eigenvalues of the transition matrix $\hat{P}$ with entries

$$\hat{P}(i,j) = p(M_i, M_j),$$

that are the probabilities of the jumps of the random walker between the sets. This matrix has the unique invariant measure $\hat{\mu}$ with $\hat{\mu}(i) = \mu(M_i)$ and is equivalent to a symmetric matrix such that its eigenvalues $1 = \hat{\lambda}_1 > \hat{\lambda}_2 \geq \ldots \geq \hat{\lambda}_m$ are real-valued again. According to the results in [12, 11] these eigenvalues approximate the dominant eigenvalues of the original transition matrix as follows:

$$E(M_1, \ldots, M_m) = \max_{i=1,\ldots,m} |\lambda_i - \hat{\lambda}_i| \leq (m-1)\,\lambda_2 \max_{i=1,\ldots,m} \delta_j^2, \tag{5}$$

where the δ_j again are the projection errors introduced in (3). Analogous results for the relative error exist also and can be found in [11]. The result (5) shows that we could also identify the optimal metastable decomposition by means of minimization of the eigenvalue error $E(M_1, \ldots, M_m)$. The interpretation is obvious: The optimal sets are the ones for which the timescales/eigenvalues of the jump process defined by the resulting jump matrix $\hat{P}$ optimally approximate the dominant timescales of the original random walk.

3.2 Fuzzy modular partition

The complete partition of the network into modules has at least one essential drawback: Since we are interested in identifying the hubs that provide the main connection between modules, we should *not* be interested in a complete partition: in a complete partition every node belongs to *exactly one* module while we are interested in nodes that *cannot* be assigned clearly to one of the modules but somehow fall in between. Thus we should be interested in finding modules $M_1, \ldots, M_m \subset V$ that are still disjoint but do not form a full partition:

$$\cup_{j=1}^m M_j \neq V \quad \Rightarrow \quad M = V \setminus \cup_{j=1}^m M_j \neq \emptyset.$$

We can imagine the M_j as the "cores" of the metastable partition sets discussed in the last section; we will therefore call them *core sets*.

The next obvious question now is: To which extent does a node $x \in M$ belong to one of the core sets $M_1, \ldots, M_m$? In other words, how much is $x \in V$ committed towards one of the core sets M_i? In order to answer this question we introduce the *committor* functions

$$q_i(x) = \mathbb{P}[\tau_x(C_i) > \tau_x(M_i)], \qquad C_i = \cup_{\substack{j=1 \\ j \neq i}}^m M_j,$$

where $\tau_x(A)$ is the first hitting time of the set $A \subset V$ by the process (X_t) if started in x, $\tau_x(A) = \inf\{t \geq 0 : X_t \in A,\, X_0 = x\}$. Therefore, $q_i(x)$ gives us the probability that the walker, if started in node x, enters the core set M_i earlier than the union C_i of the other core sets. Despite its seemingly rather abstract definition, the function $q_i : V \to [0,1]$ can easily be computed by solving the following linear problem [33, 12, 11]

$$\begin{aligned} (\mathrm{Id} - P)q_i(x) &= 0, \quad \forall x \in M \\ q_i(x) &= 1, \quad \forall x \in M_i \\ q_i(x) &= 0, \quad \forall x \in C_i \end{aligned} \tag{6}$$

Fortunately we can easily see that the committor functions $q_1, \ldots, q_m$ form a partition of unity,

$$\sum_{i=1}^m q_i(x) = 1, \quad \forall x \in V,$$

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)

such that we can interpret $q_i(x)$ as the natural walker-based probability of assignment of node x to core set M_i.

Let us compare this case to the complete partition case. In the latter case we have the partition of unity $\sum_i \mathbf{1}_{M_i} = 1$ resulting from the "crisp" assignment probabilities $\mathbf{1}_i(x)$ of node x to core set M_i and the jump matrix $\hat{P}$ between the sets results from Galerkin projection of the original transition matrix onto the span of the $\mathbf{1}_i$. In the former case, we have the partition of unity $\sum_i q_i = 1$ resulting from the "fuzzy" assignment probabilities $q_i(x)$ of node x to core set M_i. Can we again get the associated jump matrix via Galerkin projection onto the space

$$D_c = \text{span}\{q_1, \ldots, q_m\}$$

spanned by the committor functions? It turns out that we can. However, in order to understand this, we should first ask how is the jump matrix defined.

For analyzing the jump dynamics of the walker (X_k) between the core sets, we introduce the *milestoning process* $(\hat{X}_t)$

$$\hat{X}_t = i \Leftrightarrow X_{\sigma(t)} \in M_i, \text{with } \sigma(t) = \sup_{s \leq t} \left\{ X_s \in \bigcup_{k=1}^{n} M_k \right\}, \tag{7}$$

i.e. the milestoning process is in state i, if the original process came last from core set M_i, cf. [17, 12, 11]. That is, we assign the walker to core set M_i (if this was the last core set visited) as long as it has not entered another core set. Defined in this way, the milestoning process represents the switching dynamics of the original process between the core sets and we can calculate its transition matrix.

Since the walker is reversible, the probability of finding the walker in x, conditional on that it last came from core set M_i, is the same as the probability of finding it in x, conditional on that it will enter M_i next. Thus, it is given by $\mu(x)q_i(x)$, again involving the committor defined above. As a consequence, we get a simple expression for the invariant measure of the milestoning process,

$$\hat{\mu}(i) = \sum_{x \in V} \mu(x) q_i(x).$$

We now compute the jump matrix $\hat{P}$ between the core sets M_i by the jump matrix that is generated by the milestoning process, i.e., compute its entries due to

$$\hat{P}(i,j) = \mathbb{P}_\mu[\hat{X}_1 = j | \hat{X}_0 = i].$$

As a result of a lengthy computation we find that [12, 11]

$$\hat{P}(i,j) = \frac{\langle q_i, P q_j \rangle_\mu}{\hat{\mu}(i)},$$

which tells us that in fact $\hat{P}$ can be understood as a Galerkin projection of P onto the space D_c spanned by the committors. This matrix again is equivalent to a symmetric matrix such that its eigenvalues $1 = \hat{\lambda}_1 > \hat{\lambda}_2 \geq \ldots \geq \hat{\lambda}_m$ are real-valued. According to the results in [12, 11] these eigenvalues again approximate the leading eigenvalues of the original transition matrix as follows:

$$E(M_1, \ldots, M_m) = \max_{i=1,\ldots,m} |\lambda_i - \hat{\lambda}_i| \leq (m-1)\, \lambda_2 \max_{i=1,\ldots,m} \delta_j^2, \tag{8}$$

as in (5), but now the δ_j are the projection errors of the dominant eigenvectors u_j of P onto D_c, that is, $\delta_j = \|(\text{Id} - Q)u_j\|_{2,\mu}$, as in (3) but this time Q denotes the orthogonal projection onto D_c.

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)

Not surprisingly, equation (8) reduces to (5) if the core sets $M_1, \dots, M_m$ form a complete partition such that $q_i = \mathbf{1}_{M_i}$ for all $i = 1, \dots, m$.

The obvious next question is how to define the *optimal* core sets $M_1, \dots, M_m$. Here we adopt the last perspective introduced in Section 3.1: The optimal core sets $M_1^*, \dots, M_m^*$ minimize the above eigenvalue error, i.e.,

$$M_1^*, \dots, M_m^* = \text{argmin}_{[M_1,\dots,M_m]} E(M_1, \dots, M_m). \tag{9}$$

3.3 Determining the number of modules

Summarizing our above discussion, the leading eigenvalues $\lambda_1, \dots, \lambda_m$ of the random walker's transition matrix enduce its main metastabilities and thus the information about the modules. In order to optimally define the modules, we should solve the minimization problem (9) and take the optimal core sets $M_1^*, \dots, M_m^*$ as the optimal modules.

The only problem we did not really address so far is the problem of how the number m of modules should be selected optimally. If there really is a *gap* in the spectrum of P, i.e., $|\lambda_{m+1}/\lambda_m| \ll 1$ for some small m, then our estimate (2) tells us that the averaged metastability of the optimal partition into m sets is much higher than that of the optimal $m+1$-partition. Then, m seems to be a more appropriate choice for the number of modules than $m+1$. However, in many practical applications we will not have such a clear gap in the spectrum. Thus choosing the number of modules is an important issue for which we still lack a solution.

Since we have already established the connection between the problem of partitioning and the diffusion map maximization, it seems natural to turn to diffusion maps again to approximate the number of modules in our network. From the definition of the diffusion distance we know that vertices that are connected by many short paths in the networks have a small diffusion distance. From this we conclude that dense modules in the network, where the vertices are connected by many short paths, will correspond to dense clusters in the diffusion maps projection in some dimension. We therefore elected to cluster the diffusion map and use the resulting number of outputted clusters as an indicator for m. To set aside the question of what dimension diffusion map best represents the clusters in the data, we opt instead to cluster the diffusion maps in increasing dimensions and choose as m the number of clusters that appears most frequently and consecutively.

In higher dimensions the points are spread in such a way that possibly overlapping clusters in lower dimensions become separable in higher dimensions. This is due to the data in high dimensions being more sparse, and as a result the local neighborhood of a point contains fewer other points. Of course, if the dimension of the data is too high the points are too spread out to comprise identifiable clusters (see curse of dimensionality). For a given number of points n, clustering in a dimension d that is too high results in most of the points becoming isolated *outliers*, points not belonging to any cluster. We therefore find clusters in the diffusion maps projections with dimension up to d where most of the points can be labelled as outliers.

We are now left with the question of which clustering algorithm to apply to the diffusion maps of different dimensions. The clustering algorithm of choice should have several desirable properties.

1. The algorithm should contain a good estimator for the number of clusters, and not simply rely on user input.

2. The algorithm should not partition the data, as many clustering algorithms aim to do, but rather detect clusters and identify the outliers, those points that do not belong to any cluster. This is in-line with our goal of fuzzy partitions.

3. The algorithm should perform well on density-based data and it should be possible to apply it efficiently to higher dimensional points in the dimensions determined above.

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)

DBscan. An algorithm that allows to meet our requirements is DBscan [16] (Density-Based Spatial Clustering of Applications with Noise). DBscan is a density based clustering algorithm, meaning that the clusters it outputs have a typical density of points which is considerably higher than outside of the clusters, while the density within the areas of noise is lower than the density in any of the clusters. For each point in a cluster, its neighborhood (whose radius is determined by a user-given parameter) has to contain at least a minimum number of points, so that the density in the neighborhood has to exceed some threshold. All the points in a cluster are *density reachable*, meaning that for every pair of points p_1, p_2 either p_2 is directly in p_1's neighborhood or there is a path of points from p_1 to p_2 such that every point is directly reachable from the previous point. This flexible definition does not restrict the shape of the clusters found, an additional advantage.

PCCA+. Another algorithm that answers most of our requirements is Robust Perron Cluster Analysis (PCCA+) [9, 10]. PCCA+ is a spectral clustering algorithm that results in a fuzzy clustering of data, i.e., all states are assigned to clusters within certain assignment probabilities. Assuming that we want to identify m clusters, for every state $i \in V$ and every cluster $k \in \{1, \ldots, m\}$, PCCA+ calculates the probability $\chi_k(i)$ that state i belongs to cluster k. Functions $\chi_k, k = 1, \ldots, m$, called membership functions, give the clustering information of the network in the sense that they decompose the complete state space into m metastable sets. Therefore, they are assumed to form a non-negative partition of unity $\sum_{k=1}^{m} \chi_k(i) = 1$, $i \in V$, and to be almost invariant under P. The goal of PCCA+ is to find a linear transformation matrix A that transforms $U = [u_1, \ldots, u_m]$, the first m dominant eigenvectors of P, into the membership functions $\chi = [\chi_1, \ldots, \chi_m]$, i.e., $\chi = AU$. In order to get the optimal clustering, A is chosen such that it maximizes the metastability functional

$$I(A; U, \mu) = \sum_{k=1}^{m} \frac{\langle \chi_k, P\chi_k \rangle_\mu}{\langle \chi_k, \mathbf{1} \rangle_\mu},$$

under the constraint that $\chi = AU$ forms a non-negative partition of unity; let us denote by I_m the maximum of the functional for given m.

Notice that PCCA+ does not automatically provide us with the exact estimate for the number of clusters. However, there are several techniques that can suggest the optimal choice m_{PCCA}, e.g., by running the algorithm for different cluster numbers m of clusters and determine

$$m_{PCCA} = \text{argmax}_m \frac{1}{m} I_m,$$

or by determining m_{PCCA} via the minimal overlap between the assignment functions χ_i; for more details see [10]. In addition, the PCCA+ algorithm also provides us with a good initial fuzzy modular partitioning by choosing

$$M_i = \{x \in V : \quad \chi_i(x) \geq \theta\}$$

as modules with some positive threshold parameter θ close to 1, and χ_i being the optimal membership function.

Determining m. The algorithm to be used in the following for determining the number m of clusters proceeds as follows: we run DBSCAN on the diffusion map in increasing dimensions, from 1 up to d in accordance with the number of vertices n. We then look at the number of modules outputted for each dimension, and look for the most popular choice that appears stable, meaning that the number is outputted when testing several consecutive dimensions. We then assign this number to m.

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)

4 Hubs and Transition States

Let us now assume that we have chosen the number of modules m appropriately and computed the optimal core sets $M_1, \dots, M_m$. In the following let us consider the case that the set of nodes not assigned to any of the core sets, $M = V \setminus \cup_i M_i$, is not empty, such that every $x \in M$ could be a candidate for being a hub. However, if there is a $x \in M$ such that $q_i(x) > \theta > 0.5$ for some threshold value θ that is close enough to 1, e.g., $\theta = 0.9$, then this node is committed to the core set M_i with an overwhelming probability. All the nodes $x \in V$ for which $q_i(x) > \theta$, i.e. elements of M_i together with nodes committed to M_i, will belong to the module

$$M_i^* = M_i \cup \{x \in V : \quad q_i(x) \geq \theta\},$$

and thus would not be considered as candidates for being a hub node. The set of candidate hubs should be

$$M^* = \{x \in V \setminus \cup_i M_i^* : \quad q_i(x) < \theta \; \forall i = 1, \dots, m\}.$$

Here we will study two different natural concepts for declaring a node $x \in M^*$ a hub. Both concepts are based on the idea that a hub should be essential for the communication between the modules. Communication is established by the random walker making transitions between the modules. Let us concentrate, for example, on transitions from module M_i^* to C_i^*, the union of the other modules. The frequency of these transitions, given for example by the expected number N of transitions in unit time, can be taken as a natural measure for the intensity of communication. Consequently, a hub would be a node that has a high frequency of transitions. A closer look uncovers that every single transition from module M_i^* to C_i^* can be characterized by the path the random walker takes from M_i^* to C_i^*. Imagine that we could know the ensemble of such transition paths of the walker which can help us to distinguish between important and less important transition paths. A hub would then be a node through which most of the important transition paths go. Thus, we have found two supposedly different ad-hoc definitions of a hub:

1. $x \in M^*$ is a hub if most of the communication on the way from M_i^* to C_i^* goes through x.
2. $x \in M^*$ is a hub if the most important transition paths between M_i^* to C_i^* go through x.

We will now explore these two possibilities by characterizing the transition rate and the importance of transition paths. Subsequently we will discuss which of the two possibilities might be superior algorithmically.

In order to describe the transition behaviour of the network, we will adopt the framework of transition path theory (TPT) that has been introduced in [14] for specific continuous state spaces and has been transferred to the discrete setting needed herein in [33]. We start with observing the n^{th} transition from M_i^* to C_i^* and we focus on the part of the path, when the process transits from M_i^* to C_i^*. More formally, the sequence of states

$$P_n = [x_n^{M_i^*}, x_n^1, \dots, x_n^k, x_n^{C_i^*}], x_n^{M_i^*} \in M_i^*, x_n^i \in M^*, x_n^{C_i^*} \in C_i^*, \tag{10}$$

is called the n^{th} reactive trajectory. The union of all such trajectories is called the *set of reactive trajectories*. Using this, we will study the rate at which the flow goes from one state to the next one. To this end, let us consider the *discrete probability current*

$$f_{xy}^{M_i^* C_i^*} = \begin{cases} \mu(x)\, q_i(x)\, p(x,y)\, (1 - q_i(y)), & \text{if } x \neq y \\ 0, & \text{otherwise} \end{cases} \tag{11}$$

that gives the average flow of reactive trajectories when going from state x to y, per time unit. Since the underlying process is reversible, the discrete probability current is conserved in every

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)

node outside the two sets

$$\sum_{x\in V} f_{xy}^{M_i^*C_i^*} = \sum_{x\in V} f_{yx}^{M_i^*C_i^*}, \quad y \in M^* \tag{12}$$

The net amount of probability current between two states, will provide us with the measure for the intensity of communication between the two states along the reactive trajectories. More formally, the *effective current* f_{xy}^+ defined by

$$f_{xy}^+ = \max\left(f_{xy}^{M_i^*C_i^*} - f_{yx}^{M_i^*C_i^*}, 0\right). \tag{13}$$

calculates the net average number of reactive trajectories per time unit, that make transitions from x to y when going from M_i^* to C_i^*. Again, one can show that if the underlying process is reversible, the effective current is conserved in each node outside the modules. Furthermore, it holds that the net amount of reactive trajectories that flow out of M_i^* is the same as the amount that flows into C_i^*

$$\sum_{x\in M_i^*, y\in V} f_{xy}^+ = \sum_{x\in V, y\in C_i^*} f_{xy}^+. \tag{14}$$

Using this we can describe the global transition behavior between two modules and measure how good the communication between them is. More formally, we consider the *transition rate*

$$k_{M_i^*C_i^*} = \sum_{x\in M_i^*, y\in V} f_{xy}^+, \tag{15}$$

that gives the average number of transitions from M_i^* to C_i^* per time unit. But how many of these transitions are going through a specific node $y \in M^*$? In order to answer this question, we consider a reaction path as a sequence of states that are visited when going from set M_i^* to set C_i^*. Defined in this way, a reaction path is a sequence $w = (x_0, \ldots, x_s)$, $s > 0$, of states such that $x_0 \in M_i^*$, $x_s \in C_i^*$,

$$x_k \in M, \quad \forall k = 1, \ldots, s-1,$$

and

$$f_{x_k,x_{k+1}}^+ > 0, \quad \forall k = 0, \ldots, s-1.$$

Then, for each state $y \in M^*$ outside the core sets, let us define the predecessor and successor sets, that contain the states directly before and after y on transition path

$$P_y = \{x \in V : \quad f_{xy}^+ > 0\}, \quad S_y = \{x \in V : \quad f_{yx}^+ > 0\}.$$

Using this, the reactive flow through a node $y \in M^*$ is given with

$$k_y = \sum_{x\in P_j} f_{xy}^+ = \sum_{x\in S_y} f_{yx}^+,$$

as the average number of reactive trajectories that go through the state y when going from set M_i^* to set C_i^*. An important property of this quantity is

$$k_y \leq k_{M_i^*C_i^*}, \quad y \in M^*. \tag{16}$$

In order to show this, let us fix $y \in M^*$ and consider the set W_y of all reaction paths that go through node y. Let $w_1, w_2, \ldots, w_h$ be a complete enumeration of W_j. It can be shown that these paths contain no cycles, so there have to be finitely many of them. Let us define r_l to be part of the reactive path w_l that starts with y (and ends in set C_i^*) and G to be the subgraph of the

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)

entire network that contains only edges and nodes that are contained in at least one of the r_l, $l = 1, \ldots, h$. G is a tree with root y and leaves b in C_i^*, for which we define

$$k_b = \sum_{x \in b} \sum_{z \in G \setminus b} f_{zx}^+.$$

Since $b \subset C_i^*$ and $G \subset S$ we have $k_b \leq k_{M_i^* C_i^*}$. Because of the local conservation of the flow and $G \setminus b \subset M^*$ we additionally have that $k_y = k_b \leq k_{M_i^* C_i^*}$.

Now, for every $y \in M^*$ the percentage of reactive trajectories going through y, out of all reactive trajectories from M_i^* to C_i^*, defines the importance of y in $M_i^* C_i^*$ trajectory and is given by

$$p_y^{M_i^* C_i^*} = \frac{k_y}{k_{M_i^* C_i^*}}. \tag{17}$$

Hence, the importance of the node y in the network is given with

$$p_y = \sum_{i=1,\ldots,m} p_y^{M_i^* C_i^*}. \tag{18}$$

In this sense, a hub is a node which has high importance rate, meaning that the most of communication goes through this node.

From (13) it may seem that the effective current is connected only to the local behavior of the network, but nevertheless it also determines how much flow can go through a path. This is because the flow on the reaction path is bounded by the minimal effective current of an edge involved in that path. The current that confines the flow is also called the *capacity* of the reaction path w

$$c(w) = \min_{k=0,\ldots,s-1} \{f_{x_k, x_{k+1}}^+\}, \quad w = (x_0, \ldots, x_s). \tag{19}$$

The edge with the minimal effective current is called the *dynamical bottleneck* of the path. In practical applications, reaction paths that have the maximal minimal current are of particular interest, since they can transport the most flow. These will be the most important reaction paths. We can also talk about the second most important paths and so on. For an algorithmic realization of how to find the important reaction paths please visit [33]. Furthermore, we can assign weights to the paths in a sense of how 'important' they are and observe the first few most important reaction paths. If we then again see a path as a sequence of states, we can also tell for each node how many of the important reaction paths go through this node and how important these paths are. Thus, for each node $y \in M^*$ we introduce the $M_i^* C_i^*$ path importance

$$s_y^{M_i^* C_i^*} = \frac{N_y}{N_{M_i^* C_i^*}}, \tag{20}$$

where $N_{M_i^* C_i^*}$ is the number of most important reaction paths that go from M_i^* to C_i^*, and N_y is the number of them passing through state y. This quantity will give us the measure for how much of the communication between M_i^* and C_i^* goes through one state. In this sense, the importance of node j in the network is

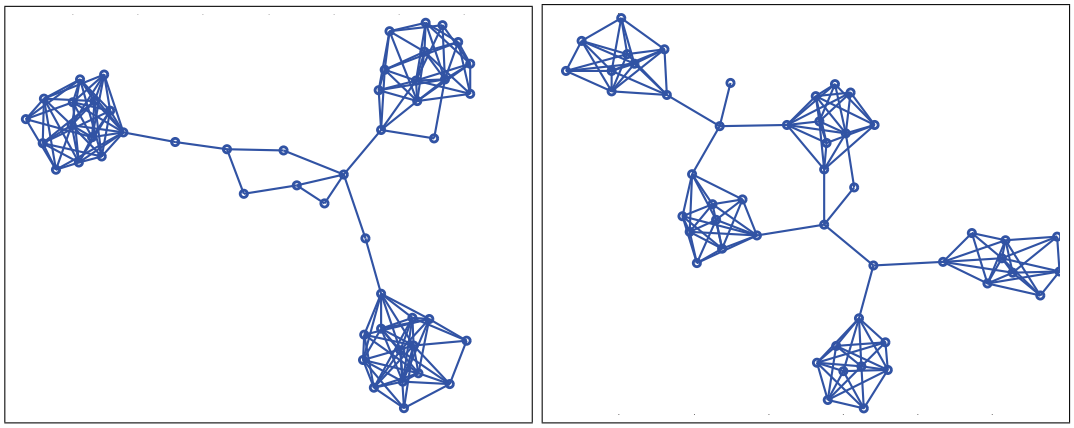
$$s_y = \sum_{i=1,\ldots,m} s_y^{M_i^* C_i^*}. \tag{21}$$

Therefore, the states that are taking part in the most intensive communications in the network are the important states hubs.

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)

5 Numerical Experiments

We now demonstrate our methods on two example networks. Network 1 consists of 50 nodes, 42 of them arranged in 3 modules, and the remaining nodes serve as intermediary nodes. Network 2 consists of 50 nodes, 45 of them arranged in 5 modules, see Figure 1. To generate the networks, we used a parameter-driven procedure that first creates dense clusters and then connects them using additional nodes.



(a) Network 1 (b) Network 2

Figure 1: The example networks, both with 50 nodes. Network 1 has three clusters, while network 2 has five.

In the first stage, we input into our procedure the number of modules k we wish to create, the total number of nodes in the modules and the desired density of edges within modules (in our case we chose $density = 0.8$). We then generate a graph for each module with these parameters using the ER (Erdos-Renyi) model [15]. In the second stage we specify the number t of intermediary nodes we would like to use to connect the modules (in our examples we chose 8 and 5 nodes, respectively) and create a set T of these new nodes. We would then like to connect the k modules and t nodes in a randomized manner. We therefore add all possible edges between pairs of nodes $(t_1, t_2), j \leq t$ in T and between every module and every intermediary node, and randomly remove a subset of these edges. To connect a module M_i to some node t_j we randomly choose a node v in M_i and construct the edge (v, t_j). The edge removal proceeds by arbitrarily ordering these new edges and removing them one after the other until the removal of some edge e disconnects the graph. Edge e is then returned to the graph to obtain a connected graph with an arbitrary connection pattern between the modules and intermediary nodes, as seen in the examples.

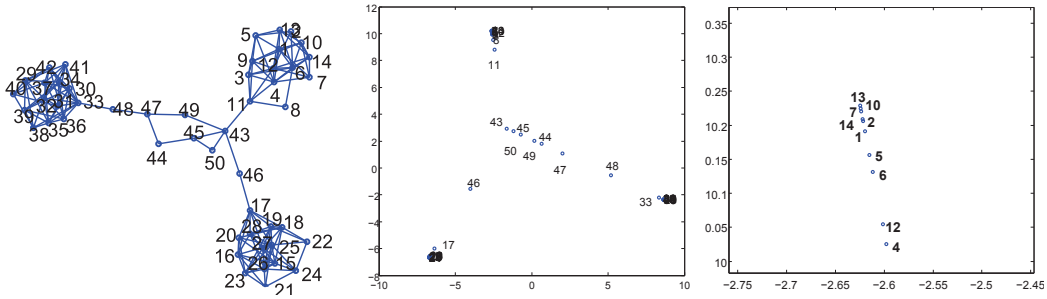
We test our methods and attempt to uncover the structure of a network as we engineered it, identifying the correct number of modules and determining the important hubs. Throughout our experiments and analysis we proceed as though we know nothing about the way the networks were constructed, not utilizing any of the parameters used in the generation process.

5.1 Estimating the number of clusters

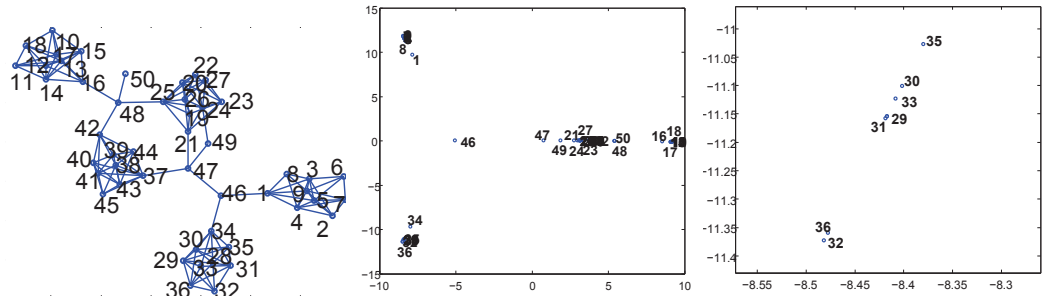
We begin by determining the number of clusters as described in Section 3.3. Recall that we estimate the number of clusters using the results of the DBscan clustering algorithm on the diffusion map

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)

in increasing dimensions. For our choice of the ϵ parameter for DBscan, we observed a *window of robustness* for every network, a range within which every choice of ϵ leads to the correct number of clusters to be outputted. From our experiments, we determine this window to be quite large. It is from this window that we sample the ϵ for each of our example networks. Figure 2 shows the $2d$ diffusion map for the two example networks, where each point is labeled with the index of the network vertex mapped to it. Note that dense areas in the diffusion maps roughly correspond to clusters already in dimension 2, as can be also seen when zooming into one of the dense areas.



(a) Network 1, labeled (b) $2d$ diffusion map (c) Close up on dense area

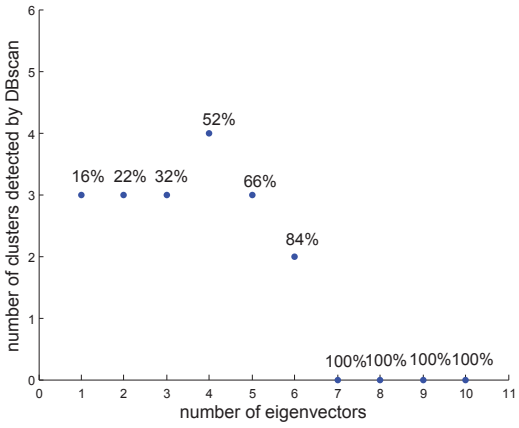


(d) network 2, labeled (e) $2d$ diffusion map (f) Close up on dense area

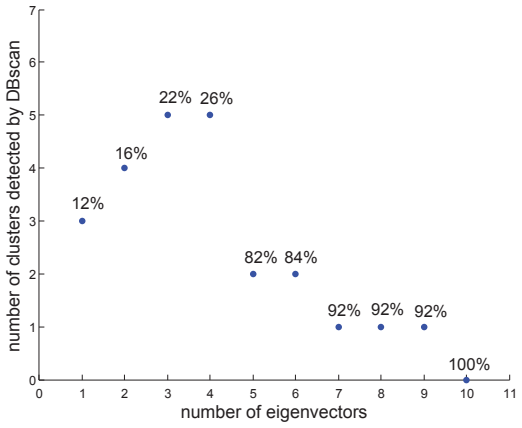
Figure 2: The labeled networks and their 2D diffusion map. Each point in the map is labeled by the index of the corresponding node. (c) and (f) shows the enlargement of a single dense area of the $2d$ diffusion map of the respective networks.

Figure 3 shows the results of running DBscan on the diffusion map of the network in increasing dimensions, giving for each dimension the number of clusters found. We use the Matlab code for DBscan, freely available from *http://code.google.com/p/dmfa07/downloads/detail?name=DBSCAN.M*, with the experimentally determined parameters $\epsilon = 0.35$ for the neighborhood radius and $k = 3$ for the minimal number of objects considered as a cluster. Looking at the number of outliers in each run of DBscan we see that indeed around dimension 10 the points are all labeled as outliers, as they are too far apart to compose clusters. We therefore focus on the first 10 dimensions, and search for a stable choice of clusters with under 90% outliers. This approach gives us an estimation of 3 and 5 clusters for the 1st and 2nd networks, respectively. Figure 4 further shows the original network again, overlayed by the clusters outputted by DBscan for those dimensions where the number of clusters was estimated correctly.

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)



(a) DBscan results for network 1. The correct solution of 3 clusters is immediately found in the first 3 dimensions.



(b) DBscan results for network 2. The correct solution of 5 clusters is found consecutively in dimensions 3 and 4.

Figure 3: Results of running DBscan on increasing dimensions, graphing the number of clusters found for every dimension. The percent of nodes marked as outliers appears next to each data point. The number of outliers increases with the dimension, finally reaching 100%.

5.2 Finding the optimal core sets and hubs

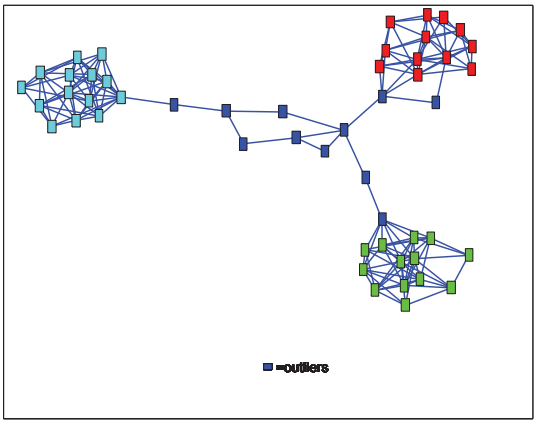
Using the centroids obtained by DBscan as the initial guess for the core sets, we can find the optimal sets defined in (9). In order to minimize (8), we apply the Simulated Annealing algorithm[5]. Figures 4(b) and 4(d) show the resulting core sets for the two networks, where elements of each core set are displayed in the same colour. As we already discussed in Section 4, there exist some nodes that are not assigned to any core set, but they belong to one core set with probability higher than θ, so they are in some sense committed to this set. Since our optimization strategy using Simmulated Annealing (SA) will find local minima in most of the cases the parameter θ is also beneficial in this case: it enriches the choice of modules with the committed nodes. Based on a variety of numerical experiments, we set $\theta \geq 0.7$ - this ensures that the solutions found (using different starting conditions for the SA) are very similar.

We also show these nodes in Figure 4(b) and Figure 4(d), marked as yellow circles, where the colour of the outline of circles correspond to the colour of the core sets to which the node is committed to. Note that these nodes are no longer considered as candidates for hubs.

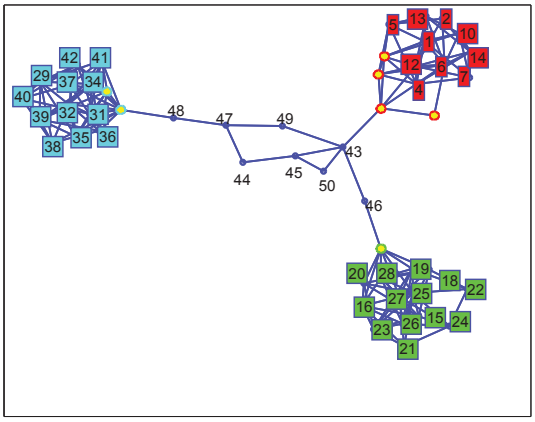
We now continue with finding the possible hubs of networks. In order to do that, we will consider the two concepts already introduced in Section 4. For every possible hub node, we calculate the importances (18) and (21). Table 5 shows the resulting importance p_y, for $y \in M^*$ for both examples. In the first network, we can see that the node 43 has the highest possible importance and indeed, this is the node through which all of the communication between all three sets is going. Nodes 46, 47 and 48 are also shown to be important, since they are the ones that connect a single core set to the others. In the second example, nodes 46, 47 and 48 have high importance and they are again the nodes that connect one set to all others.

[5]We used Matlab's standard Simulated Annealing (SA) code with standard parameters. We could of course use other heuristic optimization methods as well (e.g. genetic algorithms, swarm optimization, ...) but given the potentially very complicated solution space, we dont expect them be beneficial over SA.

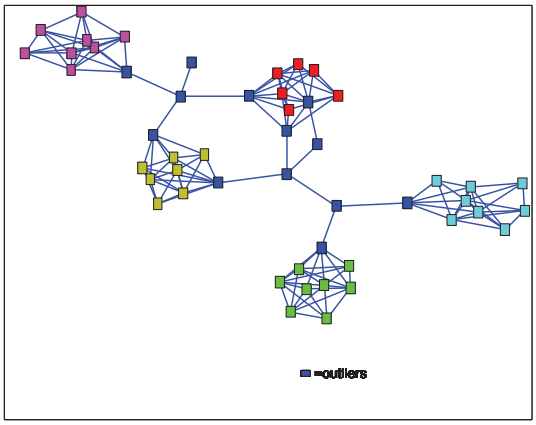
© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)



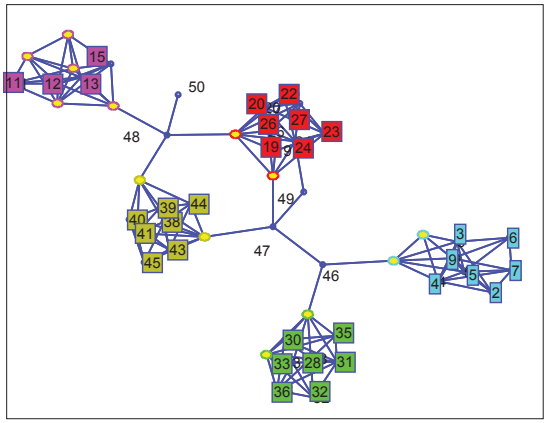
(a) DBscan clusters for network 1, dim 2.



(b) The 3 optimal core sets for network 1 and the nodes committed to these core sets.



(c) DBscan clusters for network 2, dim 4.



(d) The 5 core sets for network 2 and the nodes committed to them.

Figure 4: Comparing the two approaches on networks 1 and 2. Figures (a) and (c) show the clusters found by DBscan with the correct estimation for the number of clusters. Figures (b) and (d) show the core sets found, along with the committed nodes, which are marked with circles. The colour of the outline of circles correspond to the colour of the core set to which the node is committed. All other labelled nodes, that are not in core sets nor committed nodes to the core sets, are possible candidates for hubs.

Table 5 also shows the results for the second approach, using the rate defined in (21). Here we considered the first 15 most important trajectories between each pair of core sets. For the first network the node 43 has the highest possible rate in this characterization as well, since all trajectories pass through this node. The second most important nodes are again 46, 47 and 48. In the second network, we again have nodes 46, 47 and 48 as the most important ones.

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)

Node y	p_y	s_y
43	3.00	3.00
44	0.7178	0.6667
45	0.7178	0.6667
46	1.9690	1.9333
47	1.6748	1.6667
48	1.6748	1.6667
49	0.9570	1.00
50	0.2393	0.20

Node y	p_y	s_y
46	2.4019	2.00
47	1.9817	2.0667
48	1.9334	2.1333
49	0.5528	0.4667
50	0.00	0.00

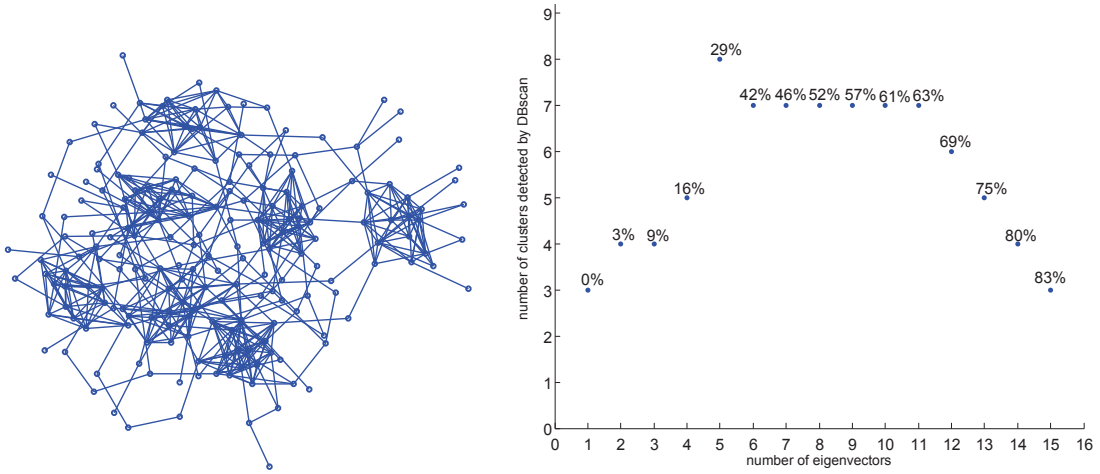
Figure 5: Rates for the hub candidate nodes for the two networks,calculated as described in Section 4. Maximal possible rate is, in both cases, the number of core sets m.

5.3 A more complex network

We have previously demonstrated the performance of our method on networks for which the structure between the modules was sparse and modules very clearly defined. We now demonstrate our analysis on a more complex network, as displayed in figure 6(a). This network has 195 nodes, 100 of them arranged in 7 modules, and is denser than our previous examples. As before, we begin by running our analysis to find the number of clusters. Our experiments show that the best assignments to clusters are obtained when taking the DBscan parameter to be $\epsilon = 5$, and the correct estimate of 7 modules is outputted already for $\epsilon = 0.65$. The chart comparing the diffusion map dimension and number of clusters is given in figure 6(b), along with the cluster assignments obtained when taking the first 7 eigenvectors for the diffusion map, shown in figure 7(a). The chart clearly indicates that the correct solution of 7 clusters is found consecutively, and overlaying the network with the cluster assignments shows that the outputted clusters are quite consistent with the modules of the original network.

We proceed, as before, by using the centroids of these clusters as the initial guess for the simulated annealing algorithm, when minimizing the eigenvalue error (8). Figure 7(b) shows the resulting 7 optimal core sets, together with the committed nodes marked as circles and coloured according to the core sets to which they are mostly committed. For all possible hub nodes j, we calculate the two rates p_y and s_y, described in Section 4. As discussed earlier, the p_y rate provides the information about how much of the total communication between 7 core sets goes through the node j. The rate s_y calculates how much of the most intensive-important communications in the network go through node j. Our algorithm results in identifying node 161 (See Figure 7(b)) as a node that has significantly higher rates than all the other possible hub nodes in this network. That means that node 161 is the node that takes part in the most of the communication and even more in the most intensive communications in the network. A close inspection of the network reveals that the node 161 is actually a direct connector for 4 core sets: 2, 5, 6 and 7, unlike any other node in this network. Clearly, our methods correctly identify the nodes most crucial to the communication between different core sets, contributing to our understanding of the underlying structure of the network.

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)



(a) Original network 3.

(b) DBscan results for network 3. The correct solution of 7 clusters is found consecutively starting in dimension 6, up until dimension 11.

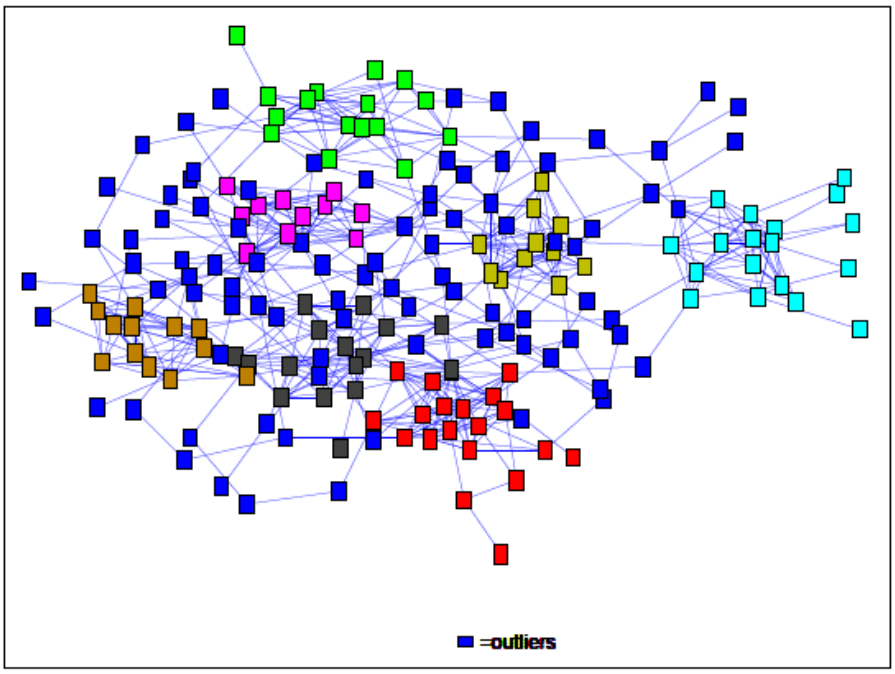
Figure 6: Network 3. A dense network and results of running DBscan algorithm for determining the number of clusters.

6 Concluding Remarks

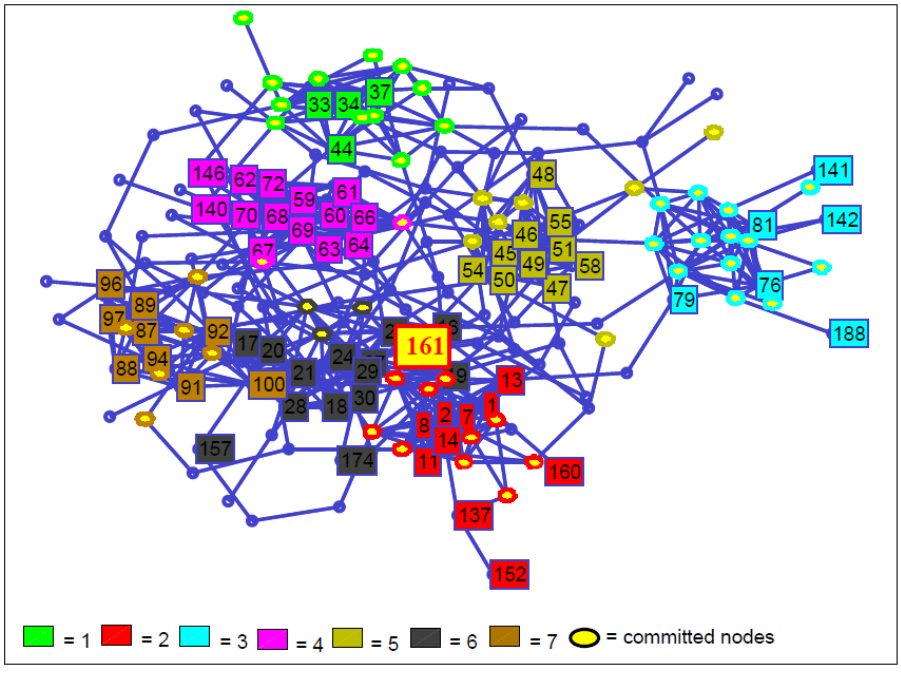
Random walks on complex networks offer a way to global analysis of the topology of the network. We have shown how to adapt recent research regarding coarse graining of random walks for the tasks of finding modules and hub states in complex modular networks. The presented algorithms are based on spectral analysis of the random walk and allow for (A) an optimal identification of fuzzy assignments of nodes to clusters/modules, (B) computation of the fraction of the overall communication between modules that goes through nodes that do not belong to any module, and (C) determination of the hubs in the network as the nodes with the highest communication load.

There are three important problems that have not be tackled. First of all, the proposed simulated annealing algorithm for identifying the optimal core sets will not perform efficiently for large networks with thousands of nodes. An efficient alternative will have to be based on analytical insight of how to improve core sets given a present core set iterate; this is subject of further investigation. Second, our above derivation of the optimality of core sets is based on the fundamental assumption that the dominant eigenvalues are positive, i.e., our approach does not cover the case (at least not automatically) in which one or several of the dominant eigenvalues are negative with large modulus. This case can typically appear if the network contains dominant ring or star structures. However, our approach can be extended to this case by replacing the transition matrix of the random walk with an appropriately designed rate matrix. This procedure will be topic of a forthcoming article. Third and lastly, we did not report on applications of our novel approach to complex real-world networks. This is mainly because there "correct" results on modules and hubs in general do not exists and we would have to compare the output of different algorithms which is a topic of its own right. Future research will demonstrate how our approach performs on, for example, biological networks in comparison to other algorithms and the insights that can be gathered through its application.

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)



(a) DBscan cluster assignment using diffusion map dimension 7.



(b) Optimal core sets with the committed nodes (marked with circles and coloured according to the colour of the core set to which they are committed).

Figure 7: Network 3. The clusters found by DBscan compared to the optimal core sets calculated by minimizing the eigenvalue error.

Acknowledgment

he authors would like to thank Marco Sarich for fruitful discussions concerning dominant negative eigenvalues and their relation to star- and loop-structures. Also, some of the preliminary work was done by James Gill, Kamron Saniee, Lara Neureither, and Bastian Kayser within the RIPS-BERLIN program sponsored by the DFG Research Center MATHEON in Berlin, and IPAM (LA, USA). ND acknowledges support by the Berlin Mathematical School (BMS).

References

[1] R. Albert and AL. Barabasi. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74(1):47–97, Jan 2002.

[2] D. Aldous and J. Fill. *Reversible Markov Chains and Random Walks on Graphs*. University of California, Berkeley, 2002.

[3] Y. Artzy-Randrup, SJ. Fleishman N. Ben-Tal, and L. Stone. Comment on "network motifs: Simple building blocks of complex networks" and "superfamilies of evolved and designed networks". *Science*, 305(5687):1107c, 2004.

[4] AL. Barabasi. *Linked: The new science of networks.* Cambridge (Massachusetts): Perseus Publishing, 2002.

[5] AL. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509:512, 1999.

[6] AL. Barabasi and ZN. Oltvai. Network biology: Understanding the cell's functional organization. *Nature Reviews Genetics*, 5:101, 2004.

[7] M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, 6:1373–1396, 2003.

[8] GE. Cho and CD. Meyer. Aggregation/disaggregation methods for nearly uncoupled Markov chains. *Technical Report NCSU no. 041600-0400, North Carolina State University*, 1999.

[9] P. Deuflhard, W. Huisinga, A. Fischer, and Ch. Schuette. Identification of almost invariant aggregates in reversible nearly uncoupled Markov chains. *Linear Algebra and its Applications*, 315:39–59, 2000.

[10] P. Deuflhard and M. Weber. Robust Perron cluster analysis in conformation dynamics. *Linear Algebra and its Applications*, 398 Special issue on matrices and mathematical biology:161–184, 2005.

[11] N. Djurdjevac, M. Sarich, and Ch. Schuette. Estimating the eigenvalue error of Markov state models. *submitted to Mult. Mod. Sim.*, 2010. Preprint download via http://www.math.fu-berlin.de/groups/biocomputing/publications/index.html.

[12] N. Djurdjevac, M. Sarich, and Ch. Schuette. On Markov state models for metastable processes. *Proceeding of the ICM 2010, Volume Invited Lecture*, 2010. Preprint download via http://www.math.fu-berlin.de/groups/biocomputing/publications/index.html.

[13] PG. Doyle and JL. Snell. Random walks and electric networks. *arXiv:math/0001057v1*, 2000.

[14] W. E and E. Vanden-Eijnden. Transition-path theory and path-finding algorithms for the study of rare events. *Annual Review of Physical Chemistry*, 61:391–420, 2010.

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)

[15] P. Erdös and A. Rényi. On Random Graphs I. *Publicationes Mathematicae Debrecen*, 6:290, 1959.

[16] M. Ester, H. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press, 1996.

[17] AK. Faradjian and R. Elber. Computing time scales from reaction coordinates by milestoning. *J. Chem. Phys.*, 120:10880–10889, 2004.

[18] D. Fell and A. Wagner. The small world of metabolism. *Nat. Biotech*, 189:1121–1122, 2000.

[19] M. Girvan and MEJ. Newman. Community structure in social and biological networks. *Proc. Natl. Acad. Sci.*, 99:78217826, 2002.

[20] R. Guimera and LN. Amaral. Functional cartography of complex metabolic networks. *Nature*, 433:895–900, 2005.

[21] LH. Hartwell, JJ. Hopfield, S. Leibler, and AW. Murray. From molecular to modular cell biology. *Nature*, 402:C47:C52, 1999.

[22] BD. Hughes. *Random walks and random environments.* Clarendon Press, Oxford, New York, 1995.

[23] H. Jeong, SP. Mason, AL. Barabasi, and ZN. Oltvai. Lethality and centrality in protein networks. *Nature*, 411:41, 2001.

[24] H. Jeong, B. Tombor, R. Albert, Z. Oltvai, and AL. Barabasi. The large-scale organization of metabolic networks. *Nature*, 407:651–654, 2000.

[25] S. Lafon and AB. Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28:1393–1403, 2006.

[26] T. Li, W. E, and E. Vanden Eijnden. Optimal partition and effective dynamics of complex networks. *Proc. Nat. Acad. Sci.*, 105, 2008.

[27] T. Li, J. Liu, and W. E. A probabilistic framework for network partition. *Phys. Rev. E*, 80, 2009.

[28] L. Lovasz. Random walks on graphs: A survey. *Bolyayi Society Mathematical Studies*, 2:1:46, 1993.

[29] I. Marek and P. Mayer. Aggregation/disaggregation iterative methods applied to Leontev and Markov chain models. *Appl. Math.*, 47, 2001.

[30] RB. Mattingly. A revised stochastic complementation algorithm for nearly completely decomposable Markov chains. *ORSA Journal on Computing*, 7(2), 1995.

[31] E. Meerbach, Ch. Schuette, and A. Fischer. Eigenvalue bounds on restrictions of reversible nearly uncoupled Markov chains. *Lin. Alg. Appl.*, 398, 2005.

[32] M. Meila and J. Shi. A random walks view of spectral segmentation. *AI and Statistics (AISTATS)*, 2001.

[33] P. Metzner, Ch. Schuette, and E. Vanden-Eijnden. Transition path theory for Markov jump processes. *Multiscale Modeling and Simulation*, 7(3):1192–1219, 2009.

© *2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)*

[34] CD. Meyer. Stochastic complementation, uncoupling Markov chains, and the theory of nearly reducible systems. *SIAM Rev.*, 31, 1989.

[35] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, and D. Chklovskii. Network motifs: Simple building blocks of complex networks. *Science*, 298:824:827, 2002.

[36] MEJ. Newman. Mixing patterns in networks. *Phys. Rev. E*, 67,:026126, 2003.

[37] MEJ. Newman. The structure and function of complex networks. *SIAM Review*, 45,:167–256, 2003.

[38] MEJ. Newman, AL Barabasi, and DJ Watts. *The Structure and Dynamics of Networks*. Princeton Univ Press, Princeton, NJ, 2006.

[39] MEJ. Newman and M Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69(2):026113, Feb 2004.

[40] JD. Noh and H. Rieger. Random walks on complex networks. *Phys. Rev. Lett.*, 92, 2004.

[41] G. Palla, I. Dernyi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435:814–818, 2005.

[42] E. Ravasz and AL. Barabasi. Hierarchical organization in complex networks. *Phys. Rev. E*, 67(2):026112, Feb 2003.

[43] E. Ravasz, AL. Somera, DA. Mongru DA, ZN. Oltvai ZN, and AL. Barabsi. Hierarchical organization of modularity in metabolic networks. *Science*, 297:1551–1555, 2000.

[44] M. Rosvall and CT. Bergstrom. Maps of random walks on complex networks reveal community structure. *Proc Natl Acad Sci*, 105:1118:1123, 2008.

[45] M. Sales-Pardo, R. Guimera, AA. Moreira, and LAN. Amaral. Extracting the hierarchical organization of complex systems. *Proc. Natl. Acad. Sci. U. S. A.*, 104:15224–15229, SEP 2007.

[46] M. Sarich, F. Noé, and Ch. Schuette. On the approximation quality of Markov state models. *Multiscale Modeling and Simulation*, 8(4):1154–1177, 2010.

[47] M. Sarich, Ch. Schuette, and E. Vanden-Eijnden. Optimal fuzzy aggregation of networks. *Multiscale Modeling and Simulation*, 8(4):1535–1561, 2010.

[48] Ch. Schuette and W. Huisinga. Biomolecular conformations can be identified as metastable sets of molecular dynamics. In *Handbook of Numerical Analysis*, pages 699–744. Elsevier, 2003.

[49] B. Tadic. Exploring complex graphs by random walks. (Editor: P. Garrido and J. Marro), *Modeling of complex systems: Seventh Granada Lectures, Granada, Spain, AIP Conference Proceedings,*, volume 661, pages 24–26. American Institute of Physics, 2002.

[50] DJ. Watts and SH. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

© 2011 European Society of Computational Methods in Sciences and Engineering (ESCMSE)