PhD Thesis

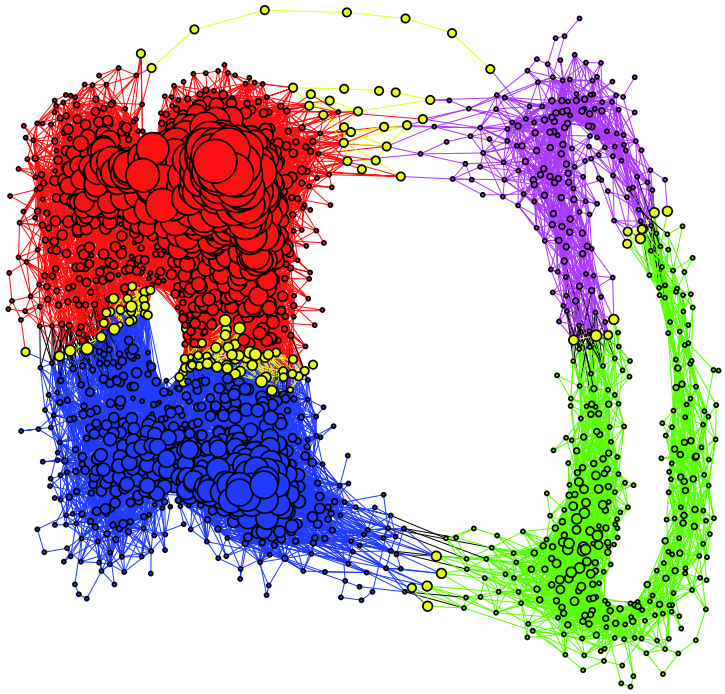
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Simplifying complex networks: from a clustering to a coarse graining strategy

David Gfeller

Laboratoire de Biophysique Statistique, ITP/SB
Ecole polytechnique fédérale de Lausanne (EPFL), 1015 Lausanne

July 6, 2007



A mes parents qui, dès mon enfance, m'ont transmis le plaisir d'apprendre de nouvelles choses...

Résumé

Le cadre mathématique des réseaux complexes s'est avéré remarquablement bien approprié pour décrire plusieurs systèmes composés d'un grand nombre d'unités qui interagissent entre elles. Chaque unité est représentée par un noeud du réseau et chaque interaction par un lien entre deux noeuds. A titre d'exemples, il est possible de cartographier les interactions entre protéines d'une même cellule ou les connections entre neurones sous forme d'un réseau. Les réseaux apparaissent aussi naturellement si l'on considère des systèmes technologiques comme internet, le WWW, ou encore le trafic aérien. Finalement les réseaux sont fréquemment utilisés en sciences sociales pour analyser les collaborations entre différents scientifiques ou les contacts humains qui sous-tendent la propagation d'une épidémie.

Pour la plupart de ces systèmes, la complexité provient principalement du grand nombre d'unités, ainsi que de la façon complexe dont elles sont interconnectées. Une approche naturelle pour simplifier de tels systèmes consiste donc à réduire leur taille. Différentes méthodes peuvent être élaborées spécifiquement pour chaque système, avec l'inconvénient que ces méthodes risquent de varier considérablement de cas en cas. D'un point de vue plus global, une alternative intéressante consiste à réduire la complexité des réseaux correspondants.

Dans cette Thèse deux stratégies sont présentées afin de réduire la complexité des réseaux. La première se réfère au paradigme bien connu du *clustering*. Le but est d'identifier des groupes de noeuds densément interconnectés et avec peu de liens vers les autres noeuds du réseau. Ces groupes sont en général appelés *clusters* ou communautés. Dans la plupart des réseaux les membres d'une même communauté ont en commun quelques caractéristiques ou propriétés qui se reflètent dans la topologie du réseau. Par exemple nous avons étudié un réseau de synonymes où les noeuds sont des mots et deux noeuds sont connectés si les mots sont synonymes. Il s'avère que les mots groupés dans une communauté représentent le plus souvent un même concept. Nous avons aussi analysé un réseau qui décrit la dynamique d'un peptide en associant un noeud à chaque configuration microscopique et un lien à chaque transition. L'étude de la structure d'un tel réseau en termes de communautés a permis d'établir une nouvelle approche pour explorer les caractéristiques principales de la dynamique du peptide et mieux comprendre la surface d'énergie libre correspondante. Finalement nous avons aussi développé une nouvelle méthode afin d'évaluer la stabilité des communautés d'un réseau en réponse à une perturbation externe du système. Cette méthode permet entre autre de quantifier à quel point les communautés reflètent réellement une organisation particulière des noeuds du réseau et non pas un simple artéfact des algorithmes utilisés pour identifier ces communautés.

Actuellement les techniques pour détecter les communautés d'un réseau sont un moyen couramment utilisé pour simplifier les réseaux, étant donné que le nombre de *clusters* est en général bien plus petit que le nombre de noeuds. Cependant une question cruciale a reçu étonnamment peu d'attention: est-ce que le réseau réduit où chaque communauté apparaît comme un seul noeud est vraiment représentatif du réseau initial? Dans cette Thèse nous avons montré que ce n'est de loin pas toujours le cas. Par exemple, nous avons comparé l'évolution d'une marche aléatoire sur le réseau initial et sur le réseau des communautés. D'importantes différences ont été observées. Afin remédier à ce problème, nous avons développé une nouvelle approche pour simplifier les réseaux complexes, tout en assurant que le nouveau (plus petit) réseau soit représentatif du réseau intial. Bien que cette approche se base aussi sur l'idée de regrouper des noeuds, comme dans le *clustering*, le but n'est plus de trouver les "vraies" communautés du réseau, mais d'obtenir un réseau réduit qui conserve les caractéristiques du réseau initial, en particulier les propriétés spectrales. De ce fait nous nous référons à cette méthode comme le *Spectral Coarse Graining*, par analogie avec le *coarse graining* utilisé en physique statistique.

En appliquant notre méthode à plusieurs sortes de réseaux, nous avons pu montrer que le réseau réduit représente une excellente approximation du réseau initial, tout en étant nettement moins grand et moins complexe. A notre connaissance, c'est la première fois qu'une telle méthode d'approximation a été élaborée dans le but de simplifer des réseaux, ce qui ouvre des perspectives extrêmement intéressantes pour étudier des grands réseaux en considérant leur version réduite.

En résumé, nous présentons tout d'abord l'utilité ainsi que les limites de l'identification des communautés d'un réseau en appliquant les méthodes existantes à différents systèmes réels représentés sous forme de réseaux. Dans une deuxième partie, nous introduisons une nouvelle startégie pour obtenir une approximation de grands réseaux par des plus petits et nous étudions plusieurs exemples d'applications afin d'illustrer l'intérêt de la méthode.

Abstract

The framework of complex networks has been shown to describe adequately a wide class of complex systems consisting of a large number of interacting units. In this framework a node is associated to each unit and two nodes are connected by an edge if the two units interact with one another. Examples of such systems can be found in living organisms as the map of interactions between proteins or the network of neurons in the brain, in artificial systems as the WWW, electrical grids or airplane connections, and in social sciences as collaboration networks or human interactions of different kinds underlying the spreading of an epidemy.

For most of these systems, the complexity arises because of the large number of units and their intricate connection patterns. A natural approach is therefore to simplify the system by decreasing its size. Different schemes can indeed be designed for each particular system, leading to effective but case-dependent methods. From a more global and statistical perspective a promising way is to reduce the complexity of the corresponding networks.

In order to simplify complex networks, two strategies are presented in this Thesis. The first approach refers to the well-known clustering paradigm. It aims at identifying groups of nodes densely connected between each other and much less to the rest of the network. Those groups are referred to as *clusters* or *communities*. For most real systems nodes within a community share some similarity or common feature. For instance we have shown that in a synonymy network where nodes are words and edges connect synonymous words, communities allowed to identify words corresponding to a single concept. We have also studied a network describing the dynamics of a peptide by associating a node to a microscopic configuration and an edge to a transition. The community structure of the network was shown to provide a new way to explore the main characteristics of the peptide dynamics and to unravel the large-scale features of the underlying free-energy landscape. Finally we have designed a new way of probing the robustness of the community structure against external perturbations of the network topology. This method allows among else to assess whether communities correspond to a real internal structure of the network or are simple artefacts of the clustering algorithms.

Community detection techniques have found a large number of practical applications as a way to simplify networks since the number of clusters is often much smaller than the number of nodes. However a crucial issue has often been disregarded: is the network of clusters truly representative of the initial one? In this Thesis we show that this is indeed not verified in many instances. For example we have considered the evolution of random walks on the network of clusters and found that it behaves quite differently than in the initial network. This observation led us to develop a new strategy to simplify complex networks, ensuring that the reduced network is representative of the initial one. It is based on the idea of merging nodes, akin to community detection. However the aim is no longer to identify the "correct" clusters, but to find a smaller network which preserves the relevant features of the initial one, and especially the spectral properties. We therefore refer to our method as *Spectral Coarse Graining*, by analogy with the coarse graining framework used in statistical physics.

Applying our method to various kinds of networks, we have shown that the coarse-grained network provides an excellent approximation of the initial one, while the size could be reduced easily by a factor of at least ten. To our knowledge it is the first time such an approximation scheme has been applied on large networks, thereby providing a well-defined way of studying large networks and their dynamics considering the much smaller coarse-grained version.

Overall, we first discuss the use and the limits of the usual clustering approach to reduce the complexity of networks, and apply it to several real-world systems. In a second part, we design a new coarse graining strategy to approximate large networks by smaller ones and provide several examples to illustrate the power of the method.

Contents

Chapter 1

Introduction

1.1 A network approach to complexity

One of the most striking features observed in several natural or artificial systems is their astonishing complexity. Even the simplest form of life relies on hundreds of intricate biochemical reactions, with the product of one reaction acting as substrate for another one, that is itself catalyzed by an enzyme generated in a third one. Larger organisms are further characterized by thousands of cells communicating with each other and the complexity becomes completely overwhelming when considering the network of neurons in our brains that consists of approximately 10^{15} to 10^{16} connections. Furthermore this complexity is not restricted to life. By now the number of web pages is estimated to roughly 10^8, with billions of links connecting them. Other examples of large and complex artificial systems include electrical grids, transportation networks or the Internet. Eventually we are all the units of a complex web of social relationships of different kinds between more than 6 billions individuals.

Since thousands of years, this complexity has fascinated people from all around the world [171] and still nowadays a complete understanding of these systems remains an open challenge. Nevertheless the examples mentioned above are characterized by an interesting common feature. They all consist of several units interacting with each other and the presence, or absence, of interactions between two units appears as one of the fundamental characteristics of the system. This property is at the origin of the use of *complex networks* (also referred to as *graphs*) as a universal and systematic framework to tackle the complexity observed in a large variety of problems. Complex networks consist of nodes connected with each other by edges. In order to describe real systems, nodes represent the basic units, being proteins, cells, web pages, cities, individuals, or anything else. Edges characterize the interactions between these units, such as protein-protein interactions, synapses connections, hyper-links, train infrastructures, social acquaintances, etc. They may have weights or directions to cope with various kinds of interactions. Indeed restricting the description of a particular system to a set of nodes and edges is a simplification and might neglect several details. But it allows to apply the same mathematical tools to various problems that originally did not have any relation, as we shall see later in this Thesis.

Historically the power of the network formalism was first noticed by the Swiss mathematician Leonhard Euler who solved the famous Königsberg bridges problem (see Fig. 1.1) and thereby became the founder of *graph theory* in 1736. The problem consisted in knowing whether it was possible to walk in the city of Königsberg and to pass by all bridges exactly once. By considering each part of land as a node and each bridge as an edge, Euler could show that this was impossible. He concluded that any graph, or network, with more than two nodes having an odd number of edges can not be fully explored without visiting more than once the same edge.

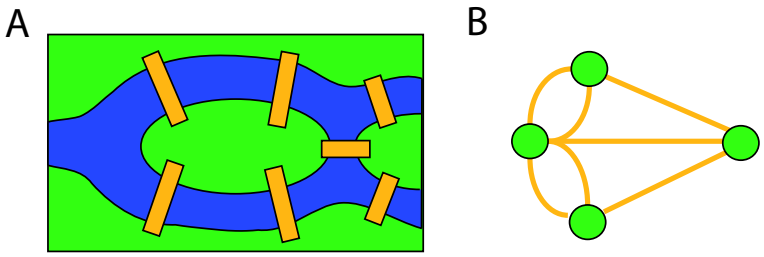


Figure 1.1: A: Schematic representation of the seven bridges of Königsberg. B: Network studied in the solution provided by Euler.

Later on, during more than two centuries, the analysis of networks was almost entirely conducted in the mathematics community and led to the seminal work of two Hungarian mathematicians, Paul Erdös and Alfréd Rényi, who laid the cornerstone of modern graph theory by defining the concept of random graphs [48] (see Chapter 2).

More recently a renewed interest in graph theory has been witnessed in computer science, social sciences and physics, triggered by the increase in computer power and the larger amount of available experimental data. In particular the tools of statistical physics, originally designed for condensed matter, proved to be extremely successful in describing several important features observed in networks. Some of these features, such as degree distribution, clustering coefficient [181], betweenness centrality [179, 119, 23], degree correlations [120], etc. have by now become standard statistical measures to characterize large complex networks. The similarities between networks stemming from various systems resulted in a vast effort to design generic models of complex networks. Two of these models, the *small-world* networks [181] and the *scale-free* networks [11], have rapidly reached a high degree of popularity, thereby stimulating a large amount of work in this field.

The success of networks in providing an adequate framework to represent and characterize complex systems hints that they may also help when dealing with the large size of these systems. Size represents an important hurdle since our minds are most often unable to grasp the main features of systems with more than a few hundred units interacting with each other. Furthermore systems whose size is larger then 10^5 become quickly impossible to deal with even with the existing computers, since useful algorithms often scale as a power of the system size, if not exponentially.

The most intuitive way to reduce the system size is to probe whether some units are similar enough so that they could be considered as a single one. This can be done "by hand". For instance in the case of the WWW, one can check whether some pages exhibit a strong similarity in their content. However this approach becomes extremely lengthy for large systems and requires a priori informations about the system units which are not always available. Instead of that, the mapping of a complex system onto a network offers the possibility to develop various statistical techniques to reduce the network size in an automated and unsupervised way, using only the network topology. For these reasons, a promising way to improve our understanding and simplify large complex systems is to reduce the complexity of the corresponding network, yet preserving its main characteristics.

1.2 Reducing the complexity

The problem of reducing the complexity of a system has its theoretical roots in statistical physics and information theory, though most applications and later developments have been found in computer science. By now, complexity reduction techniques are part of everyday life,

for instance in data compression.

In the field of complex networks and graph theory, the complexity arises mostly because of the large number of nodes and edges of real networks. Large size has several consequences. The most dramatic one is that it strongly restricts the use of time-consuming algorithms, which often hampers a detailed analysis of large networks. It further results in a very large amount of information about the system, encoded as edges between nodes, without providing a way to organize this information. Finally the graphical visualization of a network becomes often unfeasible if the number of nodes is larger than a few hundreds. To overcome these problems, two main strategies have been designed.

Historically, the first procedure that can be related with a reduction of network complexity is found in statistical physics and concerns regular networks such as periodic lattices. Taking advantage of the structure of the lattice, it has been observed that several systems described by a Hamiltonian could be mapped onto smaller ones by removing or merging a fraction of the nodes without changing the partition function Z. This process is referred to as *coarse graining*, since it provides a coarse-grained version of a system that behaves as the initial one. Systems invariant under coarse graining are said to be renormalizable and several crucial consequences follow from this property. Unfortunately, real complex networks are characterized by a much higher degree of heterogeneity than regular lattices. Therefore the requirement of invariance under coarse graining had to be abandoned and a more pragmatic approach to simplify networks has often been followed.

This approach is based on the observation that the heterogeneity of complex networks often results in a non-uniform distribution of edges between nodes. For instance social networks are characterized by groups of individuals that know each other quite well. With the goal of reducing the complexity, all these individuals may be considered as one single group. By identifying groups of nodes with a high density of edges between them, we are left with a system whose size is much smaller than the initial one. The presence of groups with a high density of edges is called the *community structure* or *cluster structure* of a network and the different techniques to identify these communities are referred to as *clustering algorithms*. The problem of identifying communities in networks was first studied in social sciences and mathematics. Later on, it has become a central topic in computer sciences with applications in parallel computing, image segmentation, database analysis or bioinformatics. For instance the function of unknown proteins could be predicted by simply considering the other members of their community in a protein interaction network [103]. More recently several clustering algorithms have been developed by statistical physicists in order to elucidate the internal organization of complex networks [66, 149, 33, 38, 137].

The two approaches mentioned above to simplify complex networks (coarse graining and clustering) differ significantly in one aspect. In the coarse graining procedure, it does not matter which nodes are grouped together, as long as the properties of the system under scrutiny are conserved. On the contrary, the clustering approach focuses on uncovering the most meaningful communities, without addressing the question whether the properties of the original network are conserved in the network of clusters. As we will show, the main novelty of this Thesis is to show that there exists a meaningful way to group nodes such that some properties of the network are preserved, as aimed by a coarse graining procedure. In addition we will present several results obtained on different kinds of real-world networks both from the point of view of their community structure and their coarse-grained structure.

1.3 Outline of the Thesis

The Thesis is divided into two main parts, accounting for the two strategies designed to simplify complex networks. Part I deals with the clustering approach and includes several applications

on different kinds of real networks. Part II presents the recently introduced coarse graining scheme, called *Spectral Coarse Graining* since it is based on the spectral properties of complex networks. On a "finer-grain" scale, the Thesis is organized in the following chapters.

Chapter 2 reviews some basic definitions, statistical measures and models of complex networks. In particular we introduce the notations that have been adopted throughout this work. For the sake of clarity, we focus on concepts which will be used extensively in the rest of the Thesis. A series of review articles and books are available for the interested readers, and references can be found in this Chapter.

In Chapter 3 the problem of uncovering the community structure of complex networks is addressed. A review of the most common clustering algorithms is provided, with an emphasis on those recently developed by scientists working in the field of complex networks.

A method to probe the robustness of the community structure of complex networks is then presented in Chapter 4. In particular we introduce the concept of *unstable* nodes, i.e nodes whose classification into a given community is ambiguous. A full characterization of unstable nodes is shown to refine the information that can be extracted from the community structure of complex networks and to assess the reliability of this information. The method is exemplified by considering a network of synonymous words in which unstable nodes turn out to be ambiguous words. Finally we show that the overall stability of the community structure can be evaluated by a single measure, the *clustering entropy*.

Chapter 5 is concerned with a particular kind of networks on which we applied the clustering and, later on, the coarse graining approach. These networks, referred to as Configuration Space Networks (CSN), describe the global architecture of the configuration space of dynamical systems by mapping each state of the system into a node and each transition into an edge. We present results obtained both for simple models of diffusive processes and for the Molecular Dynamics (MD) simulation of a di-alanine peptide. In particular the community structure of CSN is shown to unveil the different free-energy basins and barriers of the underlying free-energy landscape.

In Chapter 6 we introduce the coarse graining paradigm and review the existing approaches to coarse grain networks.

Chapter 7 describes the second strategy to simplify complex networks: the Spectral Coarse Graining of complex networks. In particular, we show that there exists an equivalence between grouping nodes in a suitable way and preserving the relevant spectral properties of complex networks. This finding opens a new, well-defined and extremely effective way to approximate large complex networks by smaller ones, and eventually to predict the properties of the initial network by considering the coarse-grained version. To fully describe the method, we first consider the coarse graining based on the stochastic matrix describing the evolution of random walks on complex networks. The coarse-grained network is shown to preserve the global dynamics, such as mean first passage times or relaxation to equilibrium, with high accuracy for various kinds of real networks. The method is further generalized to symmetric matrices, such as the adjacency matrix or the Laplacian. As an application, we consider the dynamics induced by the Gaussian Network Model [52]. In this case we show that the slow modes corresponding to fluctuations from equilibrium position are well preserved. Gaussian Networks have been used to describe the dynamics of proteins and our finding provides a mathematical framework to coarse grain large proteins based on their dynamics, rather than on their structure. At the end of the Chapter, we outline different directions in which Spectral Coarse Graining might be applied in the future.

Finally general conclusions and future perspectives of the work presented in this Thesis are exposed in Chapter 8.

Chapter 2

Basic Definitions

Describing, characterizing, and eventually clustering or coarse graining complex networks requires introducing a few statistical measures, as well as some mathematical properties of networks. Most of these concepts are borrowed from graph theory, but the recent interest in complex networks from a wider range of researchers (and particularly from statistical physicists) has resulted in a large set of tools, measures, models or algorithmic procedures. Instead of providing an exhaustive list of the recent developments in the field of complex networks, only those necessary for the understanding of the rest of this Thesis are described in this Chapter. Excellent review articles [2, 40, 121, 18, 97] or books [22, 41, 141, 129] can provide the interested reader with a more complete view of the subject.

2.1 Statistical measures

2.1.1 Networks, degree, weight and direction

A complex network is defined as a set of N nodes (representing the entities of the system under study) connected between each other by a set of M edges. This definition is equivalent to the one of a graph in the mathematical or computer science literature (actually the word "complex network" is simply aimed at enhancing that the graph represents an existing, often complex system). In these fields, nodes are also referred to as vertices, and edges as links. In this Thesis, we adopted the notation most frequently encountered in the physics community. The use of

As an alternative to enumerating the list of edges, a network is frequently described by its adjacency matrix A, where A_{ij} characterizes the connection from node j to node i. In general $A_{ij} \neq 0$ indicates the presence of an edge, while $A_{ij} = 0$ stands for the absence of edges. Different kinds of networks can be distinguished according to the value of A_{ij}:

- *Simple networks* have symmetric and binary connections: either two nodes are connected or not. The adjacency matrix is made of 1's or 0's. It is indeed a symmetric matrix. The *degree* of a node i is defined as the number of edges connected to i, $k_i = \sum_{j=1}^{N} A_{ij}$.

- *Weighted symmetric networks* allow for different values of the edge weight w_{ij}, accounting for the variability in the nature of connections between nodes. For weighted networks, the adjacency matrix takes positive values, $A_{ij} = w_{ij} \geq 0$, where $A_{ij} = 0$ indicates the absence of edges between nodes i and j. A is still symmetric. The weight of a node (also called the strength of a node [14]) is given by the sum of the weights of its edges, $w_i = \sum_{j=1}^{N} A_{ij}$. In a weighted symmetric network the degree k_i of a node i is naturally defined as the number of connections a node has, without including the weight of the connections. Although less informative than the weight, k_i indicates the total number of

nodes that interact with node i. If A_{ij}s are natural numbers, the network is a *multi-graph* and weights are interpreted as multiple edges, each of them with weight 1. The case of negative weights is not treated in this work.

- *Directed networks* are necessary to consider if the connections are not symmetric between nodes. Typically if nodes are defined as Web pages and connections as hyper-links between them, the network is clearly directed. For directed networks A is not symmetric. A directed network is unweighted if A_{ij} is either 1 or 0, and weighted if the edges weight takes any positive value. In the case of directed unweighted networks, the incoming degree of a node is given by $k_i^{\text{in}} = \sum_{j=1}^{N} A_{ij}$ and the outgoing degree is $k_i^{\text{out}} = \sum_{j=1}^{N} A_{ji}$. The generalization to directed weighted network is straightforward.

Figures 2.1 provides examples of the three cases mentioned above.

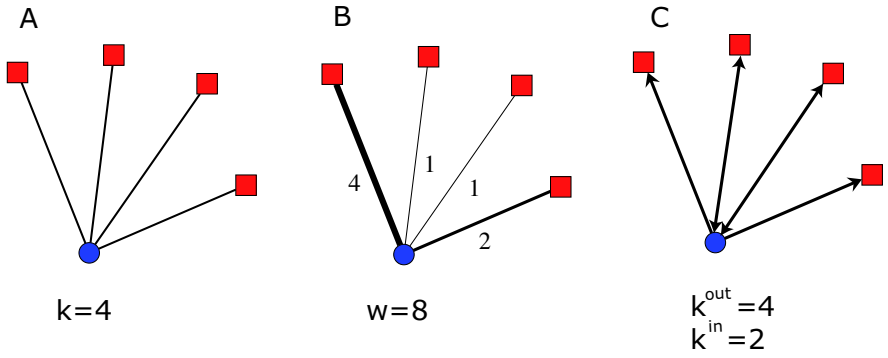


Figure 2.1: **A**: Simple network, **B**: weighted symmetric network, **C**: directed network. The degree (resp. the weight) of the blue node is indicated below.

Usually networks do not have self-edges (or loops), i.e. $A_{ii} = 0$. But even if they do the notions introduced above are perfectly suited with the only difference that a node can be neighbor of itself.

In the absence of self-edges the maximal number of edges is $M_{\max} = \frac{N(N-1)}{2}$ for undirected networks. For directed networks, one often distinguishes between the edge from i to j and the one from j to i, leading to $M_{\max} = N(N-1)$.

Finally networks whose effective number of edges M scales as N^2 are called *dense* networks while networks with M scaling as N are called *sparse* networks. In general large real networks are sparse.

2.1.2 Degree and weight distribution

While the degree of a node is a local quantity characterizing the number of connections a node has, the distribution of the degrees over the entire network is a global measure which has been extensively used to characterize networks [11]. If $N(k)$ is the number of nodes with degree k in a network, the degree distribution is $P(k) = N(k)/N$. Similarly for weighted networks the weight distribution is defined as: $P(w) = N(w)/N$. Historically the degree distribution played a central role in the study of complex networks, mainly because of the work of Barabási and Albert [11], which generated an enormous interest in characterizing complex networks by their degree distribution (see Section 2.2).

The average degree of a network is given by $< k >= \sum_{k=1}^{k_{\max}} kP(k)$ and the fluctuations of the degree around the mean values reads

$$< k^2 > - < k >^2 = \sum_{k=1}^{k_{\max}} k^2 P(k) - \left(\sum_{k=1}^{k_{\max}} k P(k) \right)^2$$

In particular in the limit of infinite networks, the fluctuations become infinite if $P(k) \sim k^{-\gamma}$ with $\gamma \leq 3$.

2.1.3 Degree correlation

An important issue to characterize complex networks is to know how nodes with high degree are connected. For instance it has been observed in social networks that nodes with large degree tend to be well connected to each other, while for several other kinds of networks nodes with large degree are statistically more connected to nodes with low degree [131]. The correlations between nodes is depicted by the average neighbor-degree,

$$k_{nn}(k) = \frac{1}{kN(k)} \sum_{i=1}^{N} \sum_{j=1}^{k_i} k_{i \to j} \delta_{k_i,k}$$

where $k_{i \to j}$ is the degree of the j^{th} neighbor of node i. Defining $P(k'|k)$ as the conditional probability that a node of degree k is connected to a node with degree k', the average neighbor-degree reads

$$k_{nn}(k) = \sum_{k'} k' P(k'|k)$$

If $k_{nn}(k)$ is flat, the network is uncorrelated. If $k_{nn}(k)$ increases with k the nodes with high degree have, on average, neighbors with high degree as well and the network is said to have a positive degree correlation, while it has a negative degree correlation if $k_{nn}(k)$ decreases with k.

Complementary to the behavior of $k_{nn}(k)$, the correlations in a network can be studied with the assortativity coefficient q [120]. Choosing randomly an edge, the degree distribution of the node at the end of this edge, not counting the edge from which we arrived, is given by:

$$e(k) = \frac{(k+1)P(k+1)}{\sum_j jP(j)}$$

Writing $e(k,k')$ for the probability distribution of the degree of the two nodes at either end of the chosen edge (still not counting this edge), the assortativity coefficient is defined as:

$$r = \frac{1}{\sigma} \sum_{k,k'} kk' \left(e(k,k') - e(k)e(k') \right) \tag{2.1}$$

with $\sigma = \sum_k k^2 e(k) - \left(\sum_k k e(k) \right)^2$ the appropriate normalization. It has been shown that $r = 0$ for perfectly uncorrelated networks. $r > 0$ (resp. $r < 0$) indicates positive (resp. negative) correlations. These two kinds of networks are often referred to as assortative (resp. disassortative) networks.

2.1.4 Clustering coefficient and cliques

By construction a node in a complex network interacts with its nearest neighbors. The number of these interactions is fully characterized by the degree of the node, but neither the degree, nor the degree correlation can measure the interactions among the neighbors of a given node. To address this question, the clustering coefficient of a node i was defined for simple networks as [181]:

$$c_i = \frac{2}{k_i(k_i-1)} \sum_{j,l=1}^{N} \frac{1}{2} A_{ij} A_{jl} A_{li}$$

The sum is equal to the total number of edges among the neighbors of node i, while $\frac{k_i(k_i-1)}{2}$ gives the maximal number of such edges (see Figure 2.2 for an example of nodes with different clustering coefficient). Averaging over the entire networks, the clustering coefficient of a network is usually defined as $c = \frac{1}{N}\sum_{i=1}^{N} c_i$ (for a slightly different definition, see [164]).

For directed unweighted network, the clustering coefficient is further given by:

$$c_i = \frac{1}{k_i^{in} k_i^{out}} \sum_{j,k=1}^{N} A_{ij} A_{jl} A_{li}$$

Akin to average neighbor connectivity $k_{nn}(k)$, the clustering coefficient as a function of the degree is defined as:

$$c(k) = \frac{1}{N(k)} \sum_{i=1}^{N} c_i \delta_{k,k_i}$$

The larger the clustering coefficient, the more connections are present between the neighbors of i and $c_i = 1$ implies that all neighbors are fully connected between each other. This naturally leads to the notion of *cliques* in a network. In a simple network, a m-clique is defined as a set of m nodes in which each node is connected to all other nodes of the set (for instance the nodes in the green circle of Figure 2.2B). If $m = 3$, cliques correspond simply to triangles. Cliques are idealized notion of communities, as it will be discussed later. If the entire network is a clique, the network is called a *complete graph*.

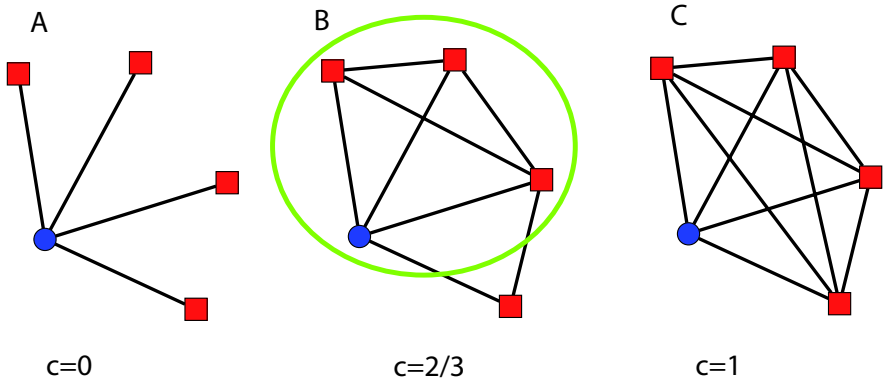


Figure 2.2: Clustering coefficient of the node represented by a blue circle. **A**: no connections are present between the neighbors, **B**: some connections are present, **C**: the neighbors are completely connected. The nodes in the green circle (B) form a clique of 4 nodes. The network of C is a complete graph.

2.1.5 Paths, connectivity and betweenness

Most pairs of nodes in a complex network are not nearest neighbors and moving from node i to node j often requires passing through several other nodes. A *path* on a network is defined as a list of nodes $a_1, a_2, ...a_l$ such that a_i has an edge pointing to a_{i+1}. The length of a path is equal to the number of steps, hence for a path including l nodes (not necessarily distinct), the

length is $l-1$. To characterize the distance between two nodes i and j, the notion of *shortest path* is crucial. The shortest path between two nodes is a path of minimal length going from i to j. It is not necessarily unique.

Using the properties of the adjacency matrix a path of length l from node j to node i exists if $\left(A^l\right)_{ji} > 0$. For unweighted networks the value of $\left(A^l\right)_{ji}$ gives the number of paths of length l from j to i.

The notion of path is immediately related to the notion of *connected* network. A network is connected if starting from any node, there exists a path to any other node. If a network is not connected a *connected component* is defined as a maximal subgraph such that there exists a path between any two nodes of the subgraph (maximal is defined here with respect to the property of connectivity and means that if any other node of the network is added to the subgraph, the subgraph is no longer connected). The case of directed network is slightly more subtle since there may exist a path from node i to node j but not vice-versa. Hence a *strongly connected component* is a maximal subgraph such that there exists a path between any two nodes of the subgraph. A *weakly connected component* is a maximal subgraph containing at least one strongly connected component C and such that for all nodes i in the subgraph there exist a path going from node i to the nodes of C. In the following a directed network is said to be connected if the entire network is a strongly connected component.

The length of the shortest path plays the role of a distance between nodes. Considering all pairs of nodes in a network, the average shortest path length is a measure of how close nodes are to each other, in other words how fast one can move on the network. For instance if nodes are connected as in a linear 1D-lattice, like in a chain, than the average distance between nodes scales as N, while for a star the average distance is constant since all nodes can be reached within 2 steps (as in Figure 2.2A). However, even if on average nodes have a small distance, some nodes might still be far from each other in a network. To account for this feature, the *diameter* of a network is defined as the longest shortest path in the network.

Finally the notion of path between nodes has also been used to characterize the centrality of nodes and edges. The most popular measure of the centrality is the *betweenness centrality*. First developed in sociology [57], the betweenness has become widely known in the community of complex network researchers with the work of Newman [119]. Intuitively a node i is central if it is part of several shortest paths. In other words removing this node would significantly modify the shortest paths between several pairs of nodes. The betweenness is defined as the sum over all pairs of nodes (j,k) of the number of shortest path between j and k that passes through node i ($b_{jk}(i)$) divided by the total number of shortest paths between j and k (b_{jk}):

$$b(i) = \sum_{\substack{(j,k) \\ j \neq k \neq i}} \frac{b_{jk}(i)}{b_{jk}} \tag{2.2}$$

Similarly the betweenness of an edge is defined as the number of shortest path between j and k that uses edge (i,m) ($b_{jk}(i,m)$) divided by the total number of shortest paths between j and k (b_{jk}):

$$b(i,m) = \sum_{\substack{(j,k) \\ j \neq k \neq i}} \frac{b_{jk}(i,m)}{b_{jk}} \tag{2.3}$$

Since computing the shortest path between any two nodes takes times $\mathcal{O}(M)$, a direct calculation of the betweenness would take times $\mathcal{O}(MN^2)$. Using breadth-first search techniques a fast algorithm was independently designed by Newman [119] and Brandes [23] that computes

the betweenness in times $\mathcal{O}(MN)$, thus allowing to compute the betweenness of large sparse networks (typically up to $N = 10'000$).

Defined as in Eq. (2.2) and Eq. (2.3) the betweenness does not include weights. An extension was designed in [122] for symmetric multi-graphs, such that if an edge has weight n, it is considered as n edges with weight 1, each of them contributing to one shortest path. If the edge weights are only required to be positive real numbers, betweenness can be generalized by means of refining the notion of distance between two neighbors. This distance is taken as a decreasing function of the edge weight (typically one chooses $d_{ij} = 1/w_{ij}$ or $d_{ij} = e^{-w_{ij}}$). The notion of shortest path is replaced by the one of optimal paths, defined as the path minimizing the distance between two nodes [140]. Using the optimal path instead of the shortest path in Eq. (2.2) and (2.3) allows to define the betweenness for weighted networks. However in the latter case the betweenness depends on the definition of the distance, and values might change significantly for different choices. Therefore the notion of betweenness appears as more appropriate for simple networks.

2.1.6 Random walks and stochastic matrices

The concept of random walks is essential to describe a large variety of phenomena (often designed as stochastic processes). In the field of complex networks, random walks are a key concept to describe diffusive processes, transport phenomena or search on a network [133]. In its discrete and simplest form the random walk moves randomly at each time step from one site (being on a lattice, a network or any other discrete system) to one of the neighbor sites.

Random walks on networks are characterized by the transition probabilities W_{ij} to move from node j to node i:

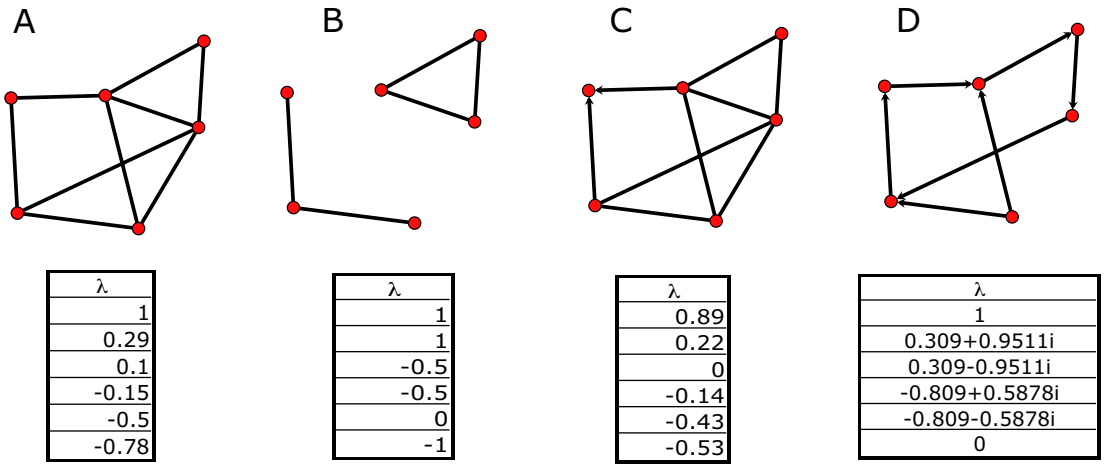
$$W_{ij} = \frac{A_{ij}}{\sum_{l=1}^{N} A_{lj}}$$

The matrix W is called the *stochastic matrix* and corresponds to the column-normalized adjacency matrix. Starting at node j, the probability to reach node i after t steps is given by $(W^t)_{ji}$. Hence the successive powers of W completely describe the evolution of a random walk on a network. If the network is connected, the evolution of random walks tends to a stationary state $|p^1\rangle$, with $W|p^1\rangle = |p^1\rangle$. For simple networks, the stationary state is proportional to the degree of the nodes since,

$$\sum_j W_{ij} k_j = \sum_j \frac{A_{ij}}{k_j} k_j = \sum_j A_{ij} = \sum_j A_{ji} = k_i \Leftrightarrow p^i \propto k^i$$

For directed (still connected) networks, the stationary state can not be trivially computed. Nevertheless very sophisticated algorithms have been designed to compute it for extremely large networks, as it was noticed that the PageRank quantity used in search engines as Google can be treated as the stationary state of a random walk over an undirected network [24, 85].

By definition, the stationary state is an eigenvector of W with eigenvalue 1. From now on we label eigenvalues such that $\lambda^1 \geq \lambda^2 \geq \ldots \lambda^N$. The Perron-Frobenius theorem ensures that if the network is connected and undirected (it can be weighted), the eigenvalue $\lambda_1 = 1$ always exists and has multiplicity 1. All other eigenvalues λ_i are real (as well as the corresponding left and right eigenvectors) with $|\lambda_i| < 1$. Other useful properties of the eigenvectors of W are discussed in Chapter 7 since they play a central role in Spectral Coarse Graining of complex networks. If the network is not connected, the Forbenius-Perron theorem does not apply. In particular there might be no stationary state (for instance if one node acts as a sink), several stationary states, or the emergence of periodic cycles corresponding to complex eigenvalues with module equal to 1 (this last situation can also occur for connected networks). Figure 2.3 shows examples of the 4 different kinds of networks and the corresponding eigenvalues λ of W.



λ
1
0.29
0.1
-0.15
-0.5
-0.78

λ
1
1
-0.5
-0.5
0
-1

λ
0.89
0.22
0
-0.14
-0.43
-0.53

λ
1
0.309+0.9511i
0.309-0.9511i
-0.809+0.5878i
-0.809-0.5878i
0

Figure 2.3: **A**: undirected connected network, **B**: undirected disconnected network, **C**: directed disconnected network with a sink, **D**: directed network with cycles.

Several global features of random walks on complex networks such as the mean first-passage time or the return probability have been studied in details [133, 13, 69, 78]. Moreover random walks have been used to characterize the centrality of an edge, leading to the definition of random walk betweenness [130, 125]. Exploration of complex networks (for instance the crawling of a search engine), traffic simulations or percolation have been described as random walks [186, 169, 79, 161, 143]. The concept of random walk has also been used to infer the community structure of networks [172, 47, 96, 95]. Finally Spectral Coarse Graining presented in Chapter 7 of this Thesis deals with an intrinsic property of stochastic matrices and makes an extensive use of random walks on complex networks.

2.2 Models of complex networks

2.2.1 Random graphs

Historically random graphs are the first generic model of networks. It was introduced by the Hungarian mathematicians P. Erdös and A. Rényi in 1959 [48] and for many decades remained the paradigm of graph theory. The model simply assumes that for each pair of nodes an edge is drawn with probability $p \leq 1$. It can be easily shown that the degree distribution is given by:

$$P(k) = \binom{N}{k} p^k (1-p)^{N-k} \approx \frac{z^k e^{-z}}{k!} \tag{2.4}$$

where $z = Np$ is the average degree. The second equality is exact only in the limit of $N \to \infty$. Eq. (2.4) shows that as k becomes larger than z, the probability to observe a node with degree k decreases exponentially with k. Random graphs are not always connected, since a very small p results in very few edges. An important result about random graphs is that if $p > \frac{1}{N}$, the size of the largest connected component scales as N, while the size of other connected components scales at most as N^α with $\alpha < 1$. Hence the largest connected component is called a "giant component".

The clustering coefficient of a random graph is given by the number of edges between the neighbors of a node, divided by the maximal number of possible edges. Since p is uniform over the network, the average clustering coefficient c is simply given by the p. Close to the critical point $p_c = \frac{1}{N}$ (which is also the condition for the graph to be sparse) $c \propto \frac{1}{N}$, and therefore decreases strongly with the size of the network.

Another important result about random graphs concerns the average shortest path length and the diameter. Both quantities were shown to scale as $\ln(N)$ [21] if $p > 1/N$, implying that even for very large graphs, the average distance between nodes remains small.

Random graphs are fascinating mathematical objects exhibiting other interesting properties than the ones mentioned above. Because of these properties, they have been used as models of real complex networks. However, despite some successes, several characteristics of real networks are not described in the framework of random graphs. In the next Sections, two features not present in random graphs, and two models providing a possible explanation for these features, are discussed. Although those models might not be relevant for all situations, they take on a historical importance as the two models that triggered a huge amount of work and shaped the actual way of looking at complex networks.

2.2.2 Small-world networks

The *small-world* model introduced by Watts and Strogatz in [181] aims at providing an explanation for two properties observed in several kinds of real networks. First, networks often have a large clustering coefficient, as expected if connections are drawn only locally (Figure 2.4A). In the same time the average distance between any two nodes is short in the sense that it does not increase proportionally to the network size. In the framework of random graphs, these two feature can only be obtained if p becomes close to one, i.e. the graph is very dense, which is not the case for most real networks.

The model of Watts and Strogatz starts from an ordered configuration of edges, as shown in Figure 2.4 A. In such a configuration, the clustering coefficient of the network is large (in the present example 0.5), but the average distance between any two nodes goes as N. Then each edge is rewired with probability p to a randomly selected node (Figure 2.4 B and C). For large p the network becomes indeed completely random, exhibiting similar properties as random graph. The transient behavior is the most interesting one. It has been shown that a small value of p is sufficient to decrease significantly the average distance between nodes, while the clustering coefficient remains almost similar to the one with $p = 0$. Hence the model displays a regime in which the clustering coefficient is large (does not decrease as $\frac{1}{N}$, as predicted in sparse random graphs), while the average distance between the nodes is small.

The word *small-world* can now be understood in terms of social networks. Most individuals on earth have the impression that their friends live in the same area and are part of the same social groups (local interactions resulting in a high clustering coefficient). However, when meeting an unknown person (for instance being abroad), it often happens that the two individuals share a common friend, or at least that some of their respective friends know each other (small-world effect). The model of Watts and Strogatz shows that only a few long-range connections are necessary to explain this phenomenon in social networks.

The presence of a large clustering coefficient and a small average distance have been found in a very large number of different networks. Although this is often an indication of a particular topology, it is not sufficient in itself to conclude that the graph is "small-world". For instance dense random graphs ($p \sim 1$) meet both criteria of small average distance and large clustering coefficient. Therefore to conclude that a network is really small-world, it is also essential to check that the network is sparse.

2.2.3 Scale-free networks

Another observation that could not be described by random graphs is the shape of the degree distribution. Random graphs show a clear exponential decay, while several real networks display a much slower decay, often better described by a power-law (also called Zipf's law), $P(k) \sim k^{-\gamma}$. Figure 2.6 shows the differences between an exponential and a power-law distribution. The

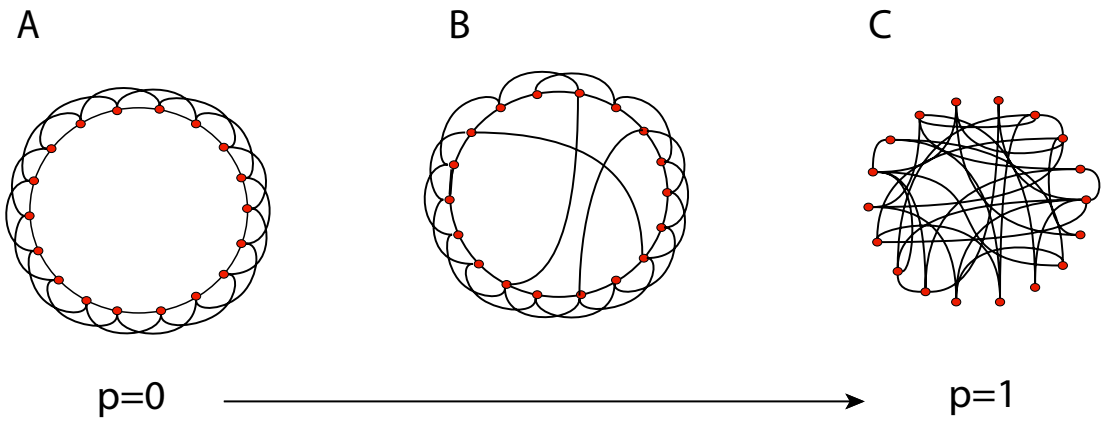


Figure 2.4: The Watts and Strogatz small-world model. **A**: $p = 0$, only nearest neighbors and next-nearest neighbors are connected. Both c and the average distance between any two nodes are large, **B**: $p << 1$ but $p > 0$, a few short-cuts are introduced in the network that tremendously reduce the average distance, without significantly altering c. **C**: p is large and the network becomes random. in this case c becomes very small.

power-law behavior has important consequences on the topology of the network, since it implies with a significant probability the existence of high-degree nodes that act as hubs in the network. Furthermore power-laws are typical of scale-free systems in statistical physics, i.e. systems having the same statistical properties at any scale. In analogy with statistical physics, the name scale-free was given to networks exhibiting a power-law degree distribution. However, because of the "small" size of complex networks ($N = 10^2 - 10^6$) compared to systems described in statistical physics ($N = N_A \approx 10^{23}$ atoms), the exact behavior of the degree distribution cannot always be computed without ambiguity. In particular speaking of a scale-free network if the number of nodes is smaller than a few hundreds is certainly meaningless.

The first models addressing the question of power-law degree distribution were discussed already long ago [183, 163, 146], but remained mostly unknown to the statistical physicists community (for a review about power-laws, see [126]). More recently Barabási and Albert (BA) introduced a highly influential model which has been a driving force behind the recent interest in networks [11]. It is a model of a growing network in which nodes are continuously added. But instead of connecting them randomly with nodes already present in the network, the connections are drawn with a probability proportional to the degree of the nodes, leading to a *preferential attachment*. In [11] a new node enters the network at each time step and connects to m existing nodes i with a probability given by:

$$P(i) = \frac{k_i}{\sum_j k_j}$$

This simply states that nodes with a high degree receive even more new connections, exemplifying the "rich-get-richer" principle. Working out the degree distribution arising from the preferential attachment rule, a power-law with exponent $\gamma = 3$ is obtained [11]. Interestingly the topology of the network displays important differences with the one obtained with random graphs. While all nodes are almost equivalent in random graphs, BA networks exhibit several hubs characterized by a much higher degree than the rest of the nodes. Figure 2.5 shows a visual comparison between random graph and BA network, and Figure 2.6 displays the different shape of the degree distribution in both cases.

The very simple model of Barabási and Albert is especially intuitive to describe the evolution of the WWW: as new pages are added to the network, they will most likely connect

to relevant and well-known pages that are already pointed by several other ones. However the exact exponent of the BA model is far from universal (for instance the WWW seems to have rather a $\gamma \approx 2.2$). To explain this discrepancy, and many others, variations of the BA model have been designed, including a non-linear preferential attachment [88], addition of edges between existing nodes [194], preferential attachment based on other quantities than the degree (typically including the effect of clustering coefficient, betweenness,...) [30, 73, 87, 54], preferential attachment including weights [14, 15, 188], and so on and so forth (the references provided here are only a short snapshot of the published works). This vast amount of work gives an idea of the exceptional popularity that this model has reached.

Another important aspect is the validity of the preferential attachment rule. Though the BA model is rather intuitive for networks like the WWW, it does not always make sense for other kinds of networks, such as protein-protein interaction networks or some social networks. Other models have been developed, aiming at providing a reasonable explanation for the origin of the degree distribution of these kinds of networks [27, 177, 173]. Such an example is presented in Chapter 5. Finally a general formalism to generate networks with power-law degree distributions with any exponent γ has been developed by Molloy and Reed [112, 113].

To conclude the short discussion about power-law distributions, the vast amount of models designed to describe features of real networks raises the question of how relevant a model is if it describes some properties of a network. In other words, how can one be sure that a model actually describes what happens in a given type of network, only by showing that some properties of the network are well described in the model. The question becomes even more delicate when noise is present in the data, such that for instance the exponent of a power-law is very difficult, if not impossible, to estimate exactly. Despite of some attempts [59], it is likely that this question has received too little attention, being diluted in the flow of enthusiasm for the new science of complex networks.

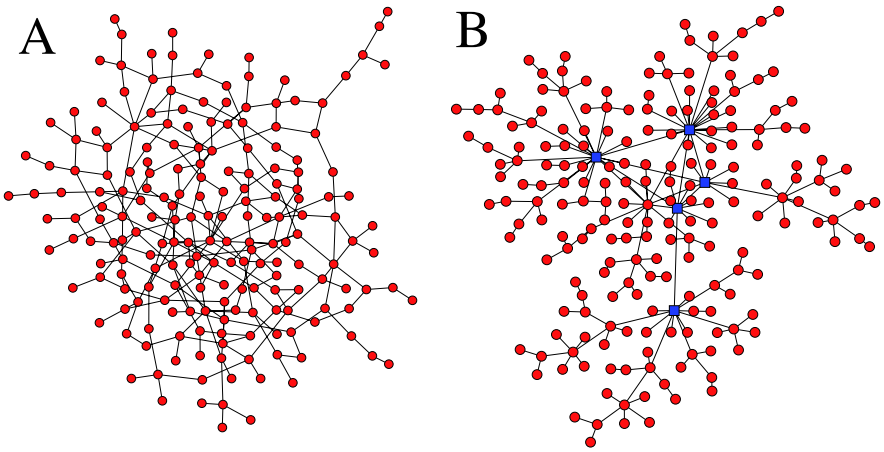


Figure 2.5: Visualization of a random graph (**A**) and a BA network (**B**). $N = 200$ and $< k >= 2$. Hubs in the BA network are represented in blue.

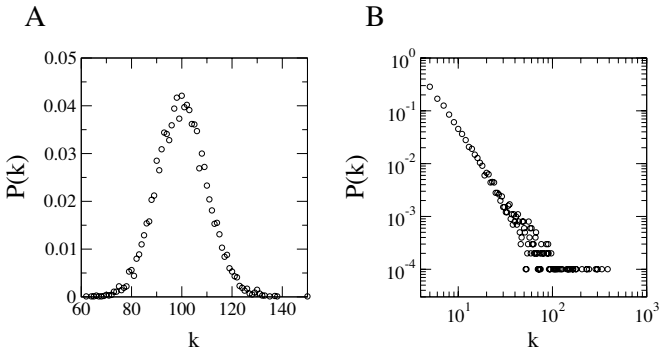


Figure 2.6: **A**: Degree distribution of a random graph of $N = 10^5$ nodes with p=0.01. **B**: Degree distribution of the BA networks with $m = 5$ and $N = 10^5$. Both networks have the same number of nodes and the same average degree $< k >= 10$.

Part I

Clustering complex networks

Chapter 3

State of the art

3.1 The paradigm of clustering

The idea underlying any clustering procedure is that the entries of a large data set can be organized and classified into a certain number of natural groups. For instance, given a list of cities from all around the world, we naturally associate each city to a country, and each country to a continent. This association process shows that, faced to a large number of data (cities), we naturally tend to use different levels of organization (countries and continents) to tackle the complexity of a system. Indeed the knowledge of the different levels of organizations has the enormous advantage of allowing to grasp the main features of the initial data set, while not including all details about each entry.

Let us now imagine that an external observer has a full knowledge of the data, but no idea of the different levels of organization. In the case of cities, he/she would know the exact position of each city on the earth and all roads connecting them, but would completely ignore their countries. Is there a way to organize the data into groups, and hopefully to recover the existing countries? Clustering techniques aim at answering positively this question.

The paradigm of clustering can therefore be stated as a *method to extract the different levels of organization of a data set by assigning each node to a particular cluster, using only the similarities or differences between nodes*. Typically information about similarity between nodes is often encoded as a network. Though stated for an abstract data set, this paradigm is extremely relevant for a vast amount of different systems. For instance cities of the same country are likely to be connected by several roads and, for most countries, are geographically close to each other. Hence if the external observer groups cities in such a way that the distances within each group are small and the road density is high, he/she will most often uncover the correct organization of the data, except for some countries such as Russia who might not fulfill the criterion of small distances between cities. In a different framework, friendships between individuals are likely to reflect the different social groups individuals take part in (clubs, professional environments, family, etc.). In this case the local information about the acquaintances of each individual may allow to unveil the second level of organization of a population. Furthermore in biology there are clear indications that by screening for instance all interactions between proteins one can infer the different groups of proteins associated with well-defined functions.

Clustering techniques represent therefore a complete shift from an analysis "by hand" to an automated approach of large data sets based only on the knowledge of the existing relations or interactions between the data. Nevertheless, a blind application of these techniques often does not allow to completely characterize the different levels of organization. For instance the external observer of the world's cities mentioned above would not gain a complete insight by identifying the correct clusters, if he/she does not know anything about the concept of country.

Similarly, the correct knowledge of the clusters of proteins is not very helpful if nothing is known about the different functions of proteins [81]. For these reasons, clustering techniques become especially useful when some partial information is known about a system. Still referring to the world's cities, if more than half of the cities in a cluster are known to belong to the same country, it is possible to infer that the remaining cities may also belong to this country, even if their country was not known *a priori*. Similarly if most proteins of a cluster have the same function, the function of the few ones that have not been annotated manually may be predicted [103]. Hence systems particularly appropriate for clustering techniques are those for which a complete information about relations between entries is available (often by experimental studies) and partial information is provided about the different levels of organization of the data. For such systems, cluster analysis allows for a more complete characterization of each entry. Furthermore this characterization can be encoded as a clustering algorithm and therefore is performed by a computer, which allows to treat very large data sets that would not be amenable for an analysis "by hand".

Nowadays clustering techniques have been applied to a very large number of problems in various fields of science and technology. However, despite several practical applications, a satisfactory mathematical definition of a cluster has never been formulated, and likely will never be. Hence the problem of finding the correct cluster structure of a data set remains a open problem for which only heuristic approaches can be designed, as we will see in the following.

3.1.1 Spatial versus graph clustering

Up to now abstract data set have been considered, without stating how interactions are represented. For instance in the example of the cities, each city is defined by coordinates representing its position on the Earth surface, whereas in the case of social networks, no distance can be defined between individuals, but friends of each individual are assumed to be known. These two examples illustrate the two main classes of data that have been analyzed in the paradigm of clustering. In the first case the data are represented in a metric space, and the interactions between data can be characterized by distances between points. This situation is referred to as *spatial clustering*. The goal of any spatial clustering algorithm is to find groups such that distances between members of the same group are small, while distances between members of different groups are large.

In the second case interactions between the entries are encoded as binary relations (i.e either two individuals interact, or not). The data are therefore best mapped onto a network. Nodes of the network represent the data (for instance individuals in social networks) and edges represent interactions (friendships). This situation is referred to as *graph clustering*. Graph clustering techniques aim at identifying the clusters that maximize the number of interactions between nodes of the same cluster, and minimize the number of interactions between nodes of different clusters.

Although the two kinds of data sets (vectors in a metric space and networks) differ by nature, the spatial clustering can always be mapped onto a graph clustering problem by defining a complete graph with edges weights as a function of the distance d_{ij} (typically $w_{ij} = \frac{1}{d_{ij}}$ or $w_{ij} = \exp(-\beta d_{ij})$). The reverse is not possible since a graph does not always imply a notion of distance between nodes (at least not in a straightforward way). For this reason graph clustering is more general and includes spatial clustering, although for many practical applications of spatial clustering there is no need to represent explicitly the data as a network. Because of this possible mapping, only the graph clustering will be discussed in details through this Thesis.

3.2 Community structure in complex networks

The study of community structure in networks has its roots in graph clustering. It is based on the observation that, contrary to the random graph model of Erdös and Rényi, real complex networks often consist of *groups in which the nodes are more connected to each other than to the rest of the network*. These groups will be referred to as *clusters*, or equivalently as *communities* in analogy with the social network terminology. The notion of community structure arises naturally for several kinds of complex networks. Typically social networks, whose nodes are individuals, are likely to have communities since we all take part in social groups (family, clubs, work,...) where people know each other quite well. While nodes represent individuals, the community structure reveals the different social groups. Elsewhere communities account for strongly interacting entities, such as web sites linked together or researchers working in the same field in a citation network. Figure 3.1 displays a typical example of a small toy-network made of 3 communities. It is evident for instance that nodes 10 and 12 belong to the same community, having exactly the same neighbors. However Figure 3.1 already raises some important questions about the definition and identification of communities. Should we consider node 14 as one single community or group it in the same community as node 8? How can we find communities in an unsupervised way for larger networks that cannot be visualized as in Figure 3.1? To which community does node 7 belong? The two first questions will be addressed in this Chapter by using the existing clustering algorithms. The last question is the core of Chapter 4, in which we present our recent developments on this subject.

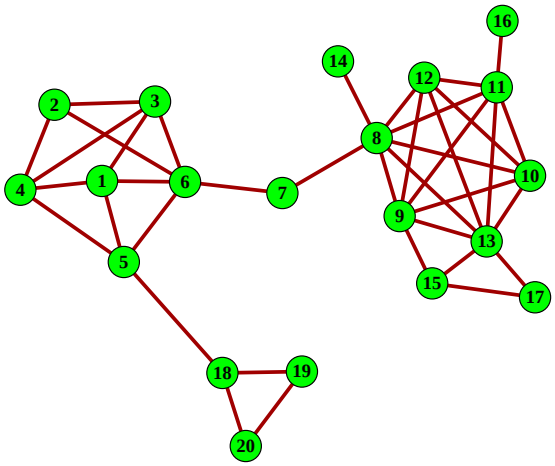


Figure 3.1: Small toy network with 3 clear clusters.

3.2.1 Evaluating the community structure of networks

Before addressing the problem of how to find the communities in a network, it is worth spending some time discussing about the notion of community. In the previous paragraph an intuitive, but highly imprecise, definition was stated: *regions in which the nodes are more connected to each other than to the rest of the network*. For instance what means "more connected"? If all nodes apart from one are grouped together, there will be many more connections within the communities than between them, but the grouping is likely to be irrelevant for most networks. A first attempt to define explicitly what is meant by *more connected to each other than to the rest of the network* is found in the bi-partitioning of networks. If S and S^c represent the two subsets of nodes, the quality of the partition could be measured by the quantity $P_1 = 1 - \frac{E(S,S^c)}{\min(|S|,|S^c|)}$, where $E(S, S^c)$ is the number of edges connecting a node of S to a node of S^c and $|S|$, resp. $|S^c|$,

is the number of nodes in S, resp. in S^c. The better the partition, the larger P_1, with a maximum at 1 if the network consists of two disconnected components. In this way the pathological case of one node grouped apart from the rest of the network does not yield a good partition since the denominator of the fraction is equal to one.

However for many real networks the split into two communities is not relevant and P_1 cannot be used as an evaluation criterion. Referring to the idea that the best communities in simple networks are cliques and that each edge between any two communities decreases the strength of the community structure, a possible measure is given by:

$$P_2 = 1 - \frac{L_{out} + \bar{L_{in}}}{N(N-1)}, \tag{3.1}$$

where L_{out} counts the number of edges connecting nodes in different communities, while $\bar{L_{in}}$ counts the number of edges that could have been placed between nodes of the same community, but do not exist. Though appealing, P_2 has several draw-backs that make it useless for large networks. In particular if a network is sparse ($M \sim N$) and if each community is made of one single node, $L_{out} \sim N$ and $\bar{L_{in}} = 0$. Hence $P_2 = 1$ in the limit of infinite networks. Several other measures of the same type as P_2 have been developed [147]. Especially interesting for evaluating the community structure is the recently introduced notion of modularity [130].

Modularity

Assuming a simple network, if l_s is the number of edges between nodes of community s and d_s is the sum of the degree of all nodes in community s, the modularity of the partition is defined as [130]:

$$Q = \sum_{s=1}^{L} \left[\frac{l_s}{M} - \left(\frac{d_s}{2M} \right)^2 \right] \tag{3.2}$$

with L the number of communities. If all nodes form one giant community, $l_1 = M$ and the modularity is 0, while if all nodes are grouped into different communities, each of size 1, $Q < 0$. The motivation to define the modularity in this way comes from the observation that if nodes are assigned completely randomly to L communities, then the expected number of edges between two different communities s and s' is given by $M \frac{d_s}{2M} \frac{d_{s'}}{2M}$. Thus modularity Q is a comparison between the real fraction of edges and the expected fraction of edges among nodes of the same community [130]. The modularity can also be expressed as a sum over all pairs of nodes [128, 150]. Writing $d_s = \sum_{i \in s} k_i$ and $l_s = \sum_{i,j \in s} A_{ij}$, Eq. (3.2) becomes

$$Q = \frac{1}{2M} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2M} \right) \delta(s_i, s_j) = \frac{1}{2M} \sum_{i,j} Q_{ij} \delta(s_i, s_j) \tag{3.3}$$

where s_i is the label of the community of node i. In this way the modularity corresponds to the Hamiltonian of a Potts model [185] with interaction strength defined by $Q_{ij} = A_{ij} - \frac{k_i k_j}{2M}$ [150]. The notion of modularity can be extended to weighted networks [122] and $Q_{ij} = w_{ij} - \frac{w_i w_j}{2W}$, where w_{ij} is the weight of the edge between node i and j, w_i is the weight of node i and $W = \sum_i w_i$ is the total weight of the network.

At this point a few comments are necessary. A random partition of a network into L communities has a modularity close to 0 and a good partition has in general a large modularity. However it should be stressed that a large Q only implies that the partition differs from a random one. In particular it does not necessarily mean that for a random network (*a priori* without any community) any partition among all the existing ones has a small modularity.

This problem was first addressed in [76]. In a more complete study [150, 151], Reichardt and Bornholdt have shown that indeed "purely random networks display intrinsic modularity and may be partitioned in a way that yields high values of Q" [151]. In their work an explicit

proof is given that modularity arises in random networks because of random fluctuations that lead, with a certain probability, to a modular structure in the network. Therefore a large value of modularity does not ensure that the network is truly modular.

As modularity became widely used, other draw-backs of this measurement have been studied. In [117], Muff and Caflisch pointed out that in the comparison between $\frac{l_s}{M}$ and $\left(\frac{d_s}{2M}\right)^2$, it is assumed that all pairs of nodes are equally likely to be connected. However for many kinds of networks, connections are allowed only within a community or between adjacent communities. In such cases the density of edges in a community should be compared to the density in the adjacent communities rather than in the whole network. With these ideas in mind, they defined the local modularity as

$$LQ = \sum_{s=1}^{L} \left[\frac{l_s}{M_s} - \left(\frac{d_s}{2M_s} \right)^2 \right],$$

where M_s is the number of edges that connect nodes of the subgraph consisting of community s and all its adjacent communities.

More recently Fortunato & Barthélémy discussed in details the resolution limits of the community detection based on modularity [53]. In particular, examples are provided where the modularity measure fails to recover the real community structure (i.e. the real community structure is not the partition with the largest modularity). In this Thesis another example of the failure of modularity to indicate the correct partition of a network is discussed in Chapter 5.

In conclusion the modularity as defined in Eq. (3.2) is a natural way of evaluating the community structure. However one should always keep in mind that high values of Q only ensures that the partition under consideration is as different as possible from a completely random assignment of the communities.

Higher order modularity

The modularity compares the presence or absence of an edge with the probability to have an edge, equal to $\frac{k_i k_j}{2M}$. However communities are not only characterized by a high density of edges. They also imply several paths of length n (n being small, $n = 2, 3$) between the nodes of a community. The expected number of paths between two nodes is given by:

$$m_{ij}^{(n)} = \sum_{l_1,\dots l_{n-1}=1}^{N} \frac{k_i k_{l_1}}{2M} \cdots \frac{k_{l_{n-1}} k_j}{2M} = \left(\frac{(N < k^2 >)}{2M} \right)^{n-1} \frac{k_i k_j}{2M}$$

To take into account paths of length $n > 1$ we introduced the n-th order modularity:

$$Q^{(n)} = \frac{1}{2M} \sum_{i,j} \left(\sum_{l=1}^{n} (A^n)_{ij} - m_{ij}^{(l)} \right) \delta(s_i, s_j)$$

Despite a theoretical interest, including higher orders in the modularity often does not improve significantly the evaluation of the community structure of complex networks, and the partition with the highest modularity is often the same with $Q^{(1)}$, $Q^{(2)}$ or $Q^{(3)}$.

3.2.2 Uncovering the community structure of complex networks

As scientists began to understand the wide range of systems described as complex networks and the importance of reducing the complexity by identifying the communities of such systems, the need for efficient and reliable clustering algorithms became evident. Both this need and the difficulty to define properly what is meant by communities or clusters have resulted in a vast amount of different procedures aimed at extracting meaningful partitions of a network. Already in the 1980's it has been reported that a questionnaire asking to describe the best clustering

algorithm and distributed to fifty-three scientists working in the field of graph clustering yielded fifty different programs and packages [89]. Only slightly exaggerating it has been said that there exist as many clustering algorithms as people doing clustering. Considering the previous discussion, no clustering algorithm can claim to be the absolute best method for all situations. Therefore a clustering algorithm should be considered as an attempt to find a partition of a network, based on an intuitive idea of what should fulfill relevant communities.

Clustering algorithms can be classified into two large classes: hierarchical and non-hierarchical. A non-hierarchical algorithm produces a unique partition of a network[1]. Non-hierarchical algorithms have the advantage of assigning each node without ambiguity to one single community. However for several networks there exist different levels of community structure. For instance in Figure 3.2A the first level consists of 3 clusters (blue circles) while the second level consists of 2 clusters (green circles). In such cases hierarchical algorithms are more convenient. The aim of hierarchical algorithms is to uncover the complete hierarchy of communities, starting from one single community up to N different ones. Results of hierarchical algorithms are often displayed as a dendrogramm where each branching corresponds to the split of a community into two smaller ones (see Figure 3.2B). Hierarchical algorithms can be further divided into two subclasses, agglomerative and divisive. Agglomerative methods start from N communities and successively merge the communities, while divisive methods start with one single community and iteratively split the communities (often by removing edges in the network) until all nodes fall into a different community. Nevertheless hierarchical algorithms cannot decide which is the best partition among the whole hierarchy (assuming that the best partition exists).

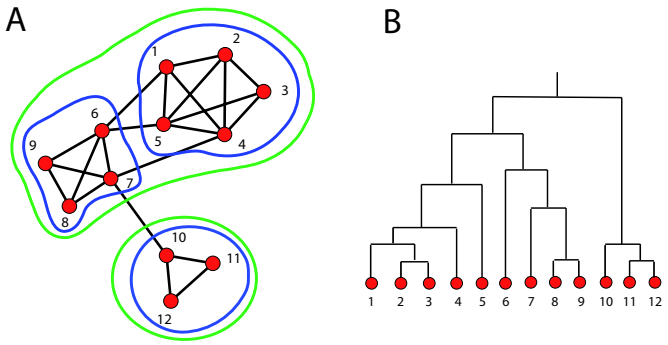


Figure 3.2: **A**: Network illustrating hierarchy in the community structure. **B**: Dendrogramm with the first, resp. the second split corresponding to groups in the green, resp. blue circles.

Having understood the two main approaches to uncover the community structure in complex networks, we will describe some of the most common clustering algorithms for complex networks, with an emphasis on those used in the analysis of real examples in this Thesis. In particular we will not discuss clustering algorithms that require to know *a priori* the number of communities such as k-means algorithms [100], since this information is most often not available for real networks.

3.2.3 The Girvan-Newman algorithm

Among all clustering algorithms developed by scientists working in complex networks, the Girvan-Newman (GN) algorithm [66] deserves to be mentioned first. Soon after its publication

[1]If the algorithm depends on one or several parameters, different communities may be obtained, but the algorithm is still non-hierarchical in the sense that once the parameters are fixed only one partition is found by the algorithm

it had already reached a very high popularity and triggered a strong interest among statistical physicists in detecting network communities. In the pas few years it has been used as a benchmark to compare most clustering algorithms developed more recently. Nevertheless, the GN algorithm is neither the most accurate, nor the fastest clustering algorithm.

The GN algorithm is a hierarchical algorithm based on edge betweenness. The betweenness of an edge counts to total number of shortest paths between any two nodes that pass through the edge. Thus an edge connecting two nodes within a densely connected region of the network has in general a low betweenness. As an extreme example, in a complete graph each edge has a betweenness equal to 1. On the other hand if an edge connects two parts of a network that are loosely connected to each other, several shortest paths between nodes in one part and nodes in the other one pass through this edge. Therefore the edge is likely to have a high betweenness. In the GN algorithm, edges are successively removed, starting from the one with the largest betweenness. As the removal is carried out, the network splits into more and more groups, until all edges have been removed. Visually the results of this divisive and hierarchical clustering are displayed in a dendrogramm, as in Fig. 3.2B.

Unfortunately a successive removal of the edges based on their betweenness calculated in the initial network was shown to perform badly [66]. In particular if two communities are connected with more than one edge, nothing can ensure that all these edges have a large betweenness. It has been shown that to obtain reasonable results it is necessary to recompute the betweenness of all edges after each removal. In summary the GN algorithm works as follows:

1. Compute the edge betweenness using the fast method described in [119, 23].
2. Remove the edge with the largest betweenness and check whether a new community has appeared. If yes, update the dendrogramm.
3. Recompute all edge betweenness.
4. Go back to 2).

As already pointed out the entire dendrogramm given by the GN algorithm does not allow to know which partition of the network should be chosen. To overcome this problem, the best partition has been defined as the partition along the dendrogramm that maximizes the modularity [130]. Therefore in its more recent version, the GN algorithm is also based on modularity. But instead of comparing the modularity for all possible partitions of the network, it only considers the N partitions along the dendrogramm, which indeed reduces considerably the complexity. The same criterion based on the maximal value of modularity has been applied for other hierarchical algorithms [38, 44].

From the point of view of time efficiency, the betweenness is a global quantity and can only be computed in times $\mathcal{O}(NM)$ (see chapter 2). Since the betweenness computation has to be carried out M times, the algorithm scales as $\mathcal{O}(NM^2)$, and therefore becomes extremely slow for large networks ($N > 10^4$). To speed up the algorithm, it has been first noticed that only the betweenness of edges belonging to the same component as the edge that has been removed need to be recomputed. To further improve the method, Tyler *et al.* [170] have shown that edge betweenness could be computed by randomly choosing a fraction of the shortest paths, thereby reducing the computational time. Finally several variations or extensions of the GN algorithm have also been designed. For instance, instead of using the edge betweenness as in the GN method, current-flow betweenness [130] or information centrality [55] have been considered.

3.2.4 Markov Clustering

In this subsection the Markov Clustering (MCL) is discussed in more details since it is the clustering algorithm most often used throughout this Thesis. MCL is based on random walks

on networks and was introduced by Stijn Van Dongen in his PhD Thesis [172]. The main idea can be summarized in the words of Van Dongen himself: *A Random Walk on a network that visits a dense cluster will likely not leave it until many of its vertices have been visited.* However the idea of performing a random walk on the network will not identify the clusters, since as time increases, the random walk ends up leaving the cluster to reach a stationary state (see Chapter 2). The idea of Van Dongen is to favor the most probable random walks, already after a small number of steps, thereby increasing the probability of staying in the initial cluster. The algorithm works as follows [47]:

1. Take the adjacency matrix A, including self-loops ($A_{ii} = 1$).
2. Normalize each column of the matrix to one, in order to obtain a stochastic matrix W.
3. Multiplication step: Multiply the matrix W by itself.
4. Inflation step: Take the r^{th} power of every element of W^2 (typically $r \approx 1.2 - 2$) and normalize each column to one to obtain a new stochastic matrix.
5. Go back to 3.

Step 3 is straightforward to understand in terms of random walks since $(W^2)_{ij}$ is the probability that a random walk starting from node j ends up in node i after 2 steps. Step 4 can be interpreted qualitatively. After normalization, the higher values on each column will get even higher compared to the other elements in the column. It means that the most probable k-step random walk starting from a node j will become even more probable compared to other random walks starting from j.

Although the convergence of the method has not been strictly proved, numerical computations show that after several iterations the matrix becomes idempotent under multiplication and inflation. Only a few lines of the matrix have non zero entries and these entries give the cluster structure of the network. Fig. 3.3 shows the result of MCL applied to the network displayed in Fig. 3.1. In Fig. 3.4, a graphical visualization of MCL evolution is displayed.

Apart from extremely rare cases, MCL converges to a matrix that gives a cluster structure without overlap. It is a parametric and deterministic algorithm in the sense that for given r, the algorithm always converges to the same matrix. In itself the parameter r tunes the granularity of the clustering. A small r corresponds to a few big clusters, whereas a big r corresponds to smaller clusters.

For practical implementation, the matrix multiplication is the time limiting step, since it takes times $\mathcal{O}(N^3)$ for complete graphs. For real networks, it is possible to take advantage of the sparse structure of W. In particular, it is very useful to introduce a cut-off at each iteration in order to keep W sparse. In this way, MCL could be applied even on large networks of $N \approx 10^4$ nodes in a reasonable time (see Chapter 4).

Interestingly the inflation can be reinterpreted in terms of an annealing procedure [157]. If W is a stochastic matrix, W^k is also stochastic. Hence the elements of W^k can always be written as:

$$(W^k)_{ij} = \frac{e^{-\beta B_{ij}^{(k)}}}{Z_j}, \tag{3.4}$$

where $Z_j = \sum_i e^{-\beta B_{ij}^{(k)}}$ and β is a real number playing the role of inverse temperature. The $B_{ij}^{(k)}$ can be interpreted as an effective potential barrier between nodes i and j. Now let $\Gamma^r(W^k)$ stand for the result of step 4 (r^{th} power + normalization):

$$(\Gamma^r(W^k))_{ij} = \frac{\left[(W^k)_{ij}\right]^r}{\sum_i \left[(W^k)_{ij}\right]^r} \tag{3.5}$$

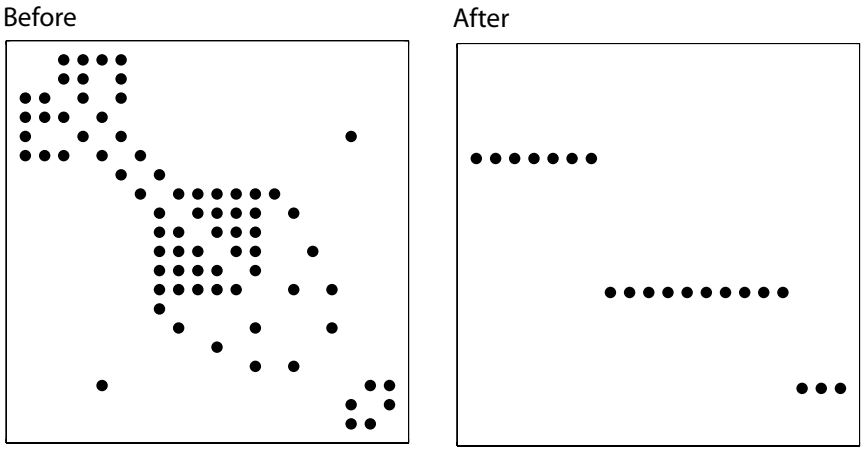


Figure 3.3: Left: Non-zero elements of the stochastic matrix W for the network in Fig. 3.1. Right: Non-zero elements of the stochastic matrix after 20 steps of MCL, $r = 1.6$.

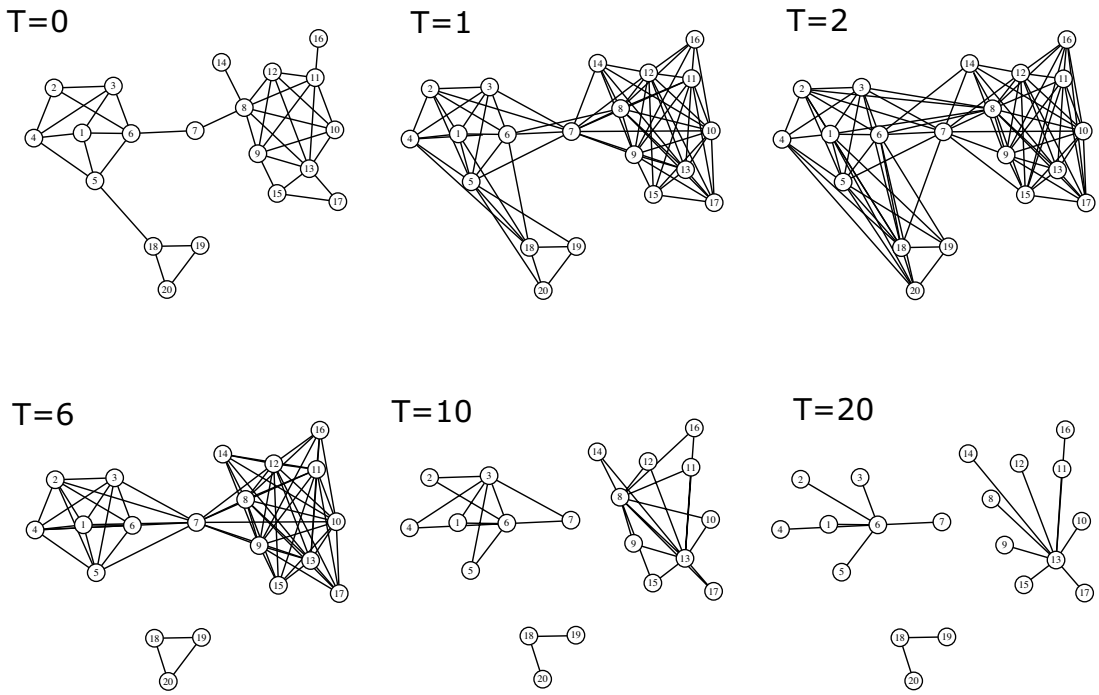


Figure 3.4: Graphical visualization of MCL evolution on the network of Figure 3.1. T gives the number of iterations. The networks are built from the elements of W larger than a cut-off of 0.01. For clarity direction on the edges has been omitted.

Using the parametrization of Eq. (3.4), we can express Γ^r as follows:

$$(\Gamma^r(W^k))_{ij} = \frac{e^{-r\beta B_{ij}^{(k)}}}{\tilde{Z}_j}$$

where:

$$\tilde{Z}_j = \sum_j e^{-r\beta B_{ij}^{(k)}}$$

Hence the inflation step corresponds to an annealing procedure of the formal temperature $T = 1/\beta$ to the lower value $T_r = T/r$. By repeatedly applying Γ^r, the system is effectively cooled down.

However, between each inflation step, the matrix multiplication changes dramatically the effective potential and makes an analytical description of the complete algorithm extremely difficult. Alternating the multiplication and the inflation is equivalent to exploring the network every time a larger number of steps (2, 2^2, 2^3, ...), but at a lower effective temperature (T, T/r, T/r^2, ...). It turns out that eventually all random walks end up at a few nodes with probability one. If two random walks starting from two different nodes (i and j) end up at the same node, i and j are classified in the same cluster.

As a final remark, a clustering technique very similar to MCL has recently been designed, using the concept of message passing [58].

3.2.5 Modularity optimization

The modularity Q was first introduced as a measure comparing the quality of different partitions of the same network [130]. Very soon it was realized that Q could be used not only to evaluate the partition into communities, but also to find the partition optimizing the modularity.

To achieve this goal, Newman designed a first agglomerative algorithm [124] (see Section 3.2.1). Initially each node is considered as one single community. Then at each step two communities are merged. The choice of the communities to be merged is done considering the largest increase in the modularity. The procedure is therefore a greedy optimization of the modularity working as a steepest descent algorithm. Considering the definition of the modularity, it has been shown that the updating of the largest increase in modularity does not require to recompute the modularity at each step [33]. Thanks to these updating rules, the algorithm can run in time $\mathcal{O}(N \ln N)$, and therefore can be used for extremely large networks (up to 10^6 nodes). Although nothing ensures that the largest modularity will be reached, there is such a large gap in computational speed between this algorithm and most others (running typically at least as $\mathcal{O}(N^2)$), that it often turns out to be a good choice (if not the only choice) for very large networks.

More involved techniques have also been used to optimize the modularity. Of particular interest is the optimization based on simulated annealing [74, 75]. The maximum of modularity is found by introducing a computational temperature T that is slowly decreased. At each step a certain amount of individual moves in which nodes change from one community to another one (as well as a few collective movements, such as merging or splitting communities) are performed. The moves are accepted with probability 1 if they result in an increase of Q and with probability $\exp(\frac{Q_f - Q_i}{T})$ if they result in a decrease of Q. Then the temperature is reduced by a factor c ($T \to cT$) with c slightly lower than 1 and a new series of randomly chosen moves are performed. The simulated annealing avoids being trapped in local maxima of the modularity, as it could be the case with the greedy optimization. The community structure found with this method has a larger modularity than with most other clustering algorithms [75], as long as the simulated annealing is run properly. Thus if modularity is the criterion chosen to define clusters, it is likely to outperform in precision any other clustering algorithms. However, the use of other clustering techniques is still important in two aspects: first speed may be a limit of the simulated annealing procedure, second for some networks modularity is known to fail in identifying the correct clusters [117, 54, 64].

3.2.6 Potts Model

The use of models directly inspired by statistical physics to elucidate the community structure of graphs goes back to the seminal work of Blatt, Wiseman and Domany [17]. A q-Potts spin

s_i is associated to each node and the dynamics is governed by a ferromagnetic Hamiltonian $H = -\sum_{<i,j>} J_{ij}\delta_{s_i,s_j}$, with J_{ij} the strength of the interaction between nodes i and j (typically $J_{ij} = A_{ij}$). At high temperature the system is in the paramagnetic phase, while at zero temperature the system is in the ferromagnetic phase. It was shown that when communities of highly connected nodes are present in the network, a superparamagnetic phase occurs: spins within the same community become perfectly correlated, while different communities remain uncorrelated. Hence the name *superparamagnetic clustering*.

As a draw-back of the superparamagnetic clustering, the choice of the "correct" temperature is not straightforward and the superparamagnetic phase is not always recognizable especially when clusters are fuzzy. For a set of data in a metric space (i.e. spatial clustering), an attempt to solve this problem was proposed by tuning the strength of the connections between the nodes to optimize the stability of the superparamagnetic phase [1]. However, in general the method, despite its elegance, has been outperformed by recent algorithms.

More recently a very nice solution to the circumvent some of the failures of the superparamagnetic clustering was proposed by Reichardt and Bornholdt [149]. Instead of considering a purely ferromagnetic Hamiltonian, they added a general anti-ferromagnetic term:

$$H = -\sum_{<i,j>} A_{ij}\delta_{s_i,s_j} + \gamma\sum_{s=1}^{q}\frac{n_s(n_s-1)}{2}$$

where q is the number of possible spins and n_s is the number of nodes with spin s. The first term on the right-hand-side favors an homogeneous distribution of spins (ferromagnetic phase). Diversity is introduced by the second sum which is minimized if all q communities have exactly the same size. The community structure is defined as the ground state of this Hamiltonian. Interestingly H can be rewritten as:

$$H = -\sum_{<i,j>} (A_{ij} - \gamma)\delta_{s_i,s_j} \tag{3.6}$$

Eq. (3.6) resembles to Eq. (3.3). Setting $\gamma = \gamma_{ij} = \frac{k_i k_j}{2M}$, Reichardt *et al.* could show that the Hamiltonian reads [150],

$$H = -\sum_{s=1}^{q}\left(l_s - \frac{d_s^2}{2M}\right) \tag{3.7}$$

where l_s is the number of edges between nodes with the same spin s and d_s is the sum of the degrees of these nodes. Eq. (3.7) is exactly the expression of the modularity Q in Eq. (3.3) multiplied by a factor $-2M$. Therefore optimizing the modularity is equivalent to minimizing the Hamiltonian of Eq. (3.7) and the problem of identifying clusters in a network has been mapped onto the one of finding the ground state of a Potts-like model.

3.2.7 Spectral Clustering

The spectral clustering of networks has a longer history than any clustering algorithm described above. It is based on the eigenvectors of the Laplacian matrix and dates back to the seminal work of Fiedler [50]. The Laplacian matrix L of an undirected network is defined as $L = D - A$, where A is the adjacency matrix and D is a diagonal matrix with $D_{ii} = \sum_j A_{ij} = k_i$ (see Chapter 2).

Eigenvectors of L have several interesting properties. For a connected network, we always have that the lowest eigenvalue $\lambda_N = 0$ and $\lambda_i > 0$, $\forall i < N$. The eigenvector p^N is equal to $(1, 1, ..., 1)$, while the sum over all other eigenvectors components is equal to 0. Eigenvector p^{N-1}, also called the Fiedler vector, is related to the optimal cut of a network, i.e. a partition of a network into two groups minimizing the number of edges between the groups. To see it, let

x be a vector such that $x_j = 1$ if node j belongs to the first group and $x_j = -1$ if not. Under this constraint, the optimal cut minimizes the following quantity:

$$\frac{1}{2}\sum_{(i,j)}(x_i - x_j)^2 A_{ij} = \frac{1}{2}\sum_{(i,j)}\left(2x_i^2 A_{ij} + 2x_i x_j A_{ij}\right) = \sum_i k_i x_i^2 - \sum_{(i,j)} x_i x_j A_{ij} = x^T L x$$

Writing $x = \sum_\alpha x^\alpha p^\alpha$, with p^α the eigenvectors of L and $\lambda^1 \geq \lambda^2 \geq \ldots \geq \lambda^N = 0$ the corresponding eigenvalues, the optimal cut is given by:

$$\min_{x_i=\pm 1} x^T L x = \min \sum_\alpha (x^\alpha)^2 \lambda^\alpha \tag{3.8}$$

The global minimum is given by $x^\alpha = 0\ \forall \alpha < N$, but does not yield a partition of the network into different clusters since $x = (1, 1, \ldots, 1)$. Hence local minima are more interesting. Unfortunately finding local minima of Eq. (3.8) turns out to be often intractable under the current conditions for x. The approach followed by Fielder is to relax the condition over x to $x^T x = 1$ and $\sum_j x_j = 0$. First we note that since $\sum_j u_j^\alpha = 0$ for $\alpha \neq N$, the condition $\sum_j x_j = 0$ is satisfied only if $\alpha_N = 0$, i.e. x is perpendicular to u^N. Then the constraint $x^T x = 1$ can be solved with the use of the Lagrange multiplier, resulting in the following equation:

$$Lx = \mu x$$

which shows that eigenvectors $\{p^1, p^2, ...p^{N-1}\}$ are extrema of $x^T L x$. It can be immediately seen from Eq. (3.8) that p^{N-1} is the global minimum under the constraint. Coming back to the initial constraints, a natural approximation of p^{N-1} is to set $x_j = 1$ if $p_j^{N-1} > 0$ and $x_j = -1$ if $p_j^{N-1} < 0$. For this reason the sign of p^{N-1} components has been used to bi-partition graphs. We further note that the constraint $x^T D x = 1$, instead of $x^T x = 1$, leads to equation

$$Lx = \mu D x \Leftrightarrow D^{-1} A x = (1 - \mu) x.$$

$D^{-1}A$ is the transpose of the stochastic matrix, called the normal matrix. Its largest eigenvalue is given by $\lambda_1 = 1$ but is associated with a right eigenvector p^1 whose components have all the same value. Remarkably, the components of other eigenvectors have the same property of summing up to zero and thus correspond to the extrema of Eq. (3.8) under the new constraint. In particular eigenvector p^2 associated with the second largest eigenvalue has similar properties as the Fiedler vector of L. For these reasons the normal matrix has also been used in graph partitioning. In Chapter 7 we provide another justification for using the normal matrix to extract the community structure of networks.

The method of Fiedler always results in a bi-partition of the network. But real networks often consist in more than two clusters and the Fiedler vector is not sufficient for extracting the community structure. A possible way around this problem is to consider each group as a new graph and apply again the bi-partition scheme based on the second eigenvector of the reduced Laplacian, as in a hierarchical approach. The method is fast since computing the second largest eigenvector can be done in a computational time $\mathcal{O}(M/(\lambda^2 - \lambda^3))$ [71, 123]. However it was shown to perform poorly for several kinds of networks.

A more interesting extension of the Fiedler method is to consider several eigenvectors of L corresponding to low eigenvalues. As a justification, it is known that if a network consists in S disconnected components, L has S eigenvalues equal to zero. The S corresponding eigenvectors can be chosen such that $p_j^{N-i} = 1$ if node j belongs to component i and 0 otherwise, with $i = 0, \ldots S-1$ labeling the S disconnected components. Now if the network has S communities weakly connected to each other, L has $S-1$ eigenvalues close to zero and much smaller than the rest of the spectrum. The $S-1$ corresponding eigenvectors reflect the community structure of the network and their components are strongly correlated within each clusters. In [28] this observation has been used to define correlations between nodes using eigenvector components.

The S lower eigenvectors of L have also been used as a way to embed a network in a metric space, associating to each node i a vector given by the S components $(p_i^{N-1}, \ldots, p_i^{N-S})$. In the embedding space, Euclidean distances can be defined between nodes and spatial clustering techniques might be used, as in [38].

More recently eigenvectors corresponding to the largest eigenvalues of the modularity matrix Q_{ij} (see Eq. (3.3)) have also been exploited in the same way to design clustering algorithms [128, 127].

3.2.8 Miscellaneous

Clique Percolation

In the strongest sense, a community on a network could be defined as a group of nodes that interact all with each other, resulting in a clique. In practice this definition of community is not relevant. For instance in a social network, an individual taking part a club does not need necessarily to interact with all other members of the club. It is enough for him to interact with a large fraction of the members to be part of the group. In addition for networks built from experimental studies, it is likely that not all connections have been recorded because of the noise inherent to experimental procedures.

Albeit cliques can not reveal directly the community structure, a clustering algorithm was built by considering the small cliques on a network [137]. Two k-cliques (cliques made of k nodes) are considered adjacent if they share $k-1$ nodes [36]. Starting from a clique, a community is defined as the union of all cliques that can be reached by rolling from one adjacent clique to another one. Apart from performing well on several kinds of networks, this method has the advantage of allowing a multiple assignation of nodes: a node can belong to more than one community if it is part of two different cliques.

Random walks

Other clustering algorithms than MCL are based on random walks onto a network. Indeed if two nodes i and j are in the same community, they tend to "see" all other nodes in the same way. Denoting by P_{il}^t the probability for a random walk starting at node i to reach node l after t steps, the last sentence translates as: $P_{il}^t \sim P_{jl}^t$. From this observation a distance between a node and a community was proposed in [96] as well as an agglomerative clustering algorithm in which communities to be merged are the ones minimizing the sum of the distances between each node and its community. Other distances between nodes of a network based on random walks have also been defined in [95] and [156] and sometimes used as a way to identify clusters. Finally Zhou [193] has used random walks to define a dissimilarity index between nodes.

Synchronization

Researchers have studied the dynamics of coupled oscillators system with a topology given by the edges of real network. In particular synchronization has received much attention [116, 115, 132]. As expected nodes within a community tend to synchronize more easily than nodes in different communities. This observation has been used to design clustering algorithms [139, 134]. For networks with well-defined communities, even the hierarchy of modules could be identified as different time-scales of the synchronization [7].

Edge-Clustering

In [147] Radicchi *et al.* have introduced the edge clustering coefficient. It is defined as the number of triangles an edge takes part in divided by the maximal possible number: $c_{ij} = \frac{z_{ij}}{\min(k_i-1, k_j-1)}$. It is likely that an edge connecting two nodes belonging to two different communities does not

take part in many triangles, while an edge connecting two nodes of the same community in general does. The algorithm of [147] successively removes edges starting from the one with the lowest c_{ji}. In this way a hierarchical divisive clustering algorithm is obtained. Since it is based on a local property of network, it has the advantage of being relatively fast (at least if the network has a local structure such that computing the clustering coefficient does not involve information about the whole network). Also based on a local quantity is the work of Eckmann and Mose [45], where the clustering coefficient of nodes is interpreted in terms of a curvature measure. Communities are inferred as regions of high curvature.

Chapter 4

Robustness of the community structure

4.1 Motivations

As seen in the previous Chapter, uncovering the community structure of networks has been the subject of a vast amount of work. In most cases clustering algorithms partition a network into non-overlapping clusters, assigning each node to a specific cluster ("hard-clustering"). While the hard-clustering has the advantage of defining without ambiguity to which community each node belongs, results are sometimes questionable. For instance situations where a node lie between two clusters are often encountered in real networks, either because the node belongs to both clusters, or because it belongs to none of them (see Figure 4.1). Furthermore, data collected from experimental procedures are often noisy and contain several false-positive or false-negative interactions, which might alter the community structure of a network. Finally it is known that applying clustering algorithms to random graphs [76, 150, 151] or to regular lattices [108] yields several communities, while there is a priori no such feature in those networks. Therefore a clustering algorithm output should always be questioned, and if possible validated.

The first way of evaluating the validity of clustering results, is to compare the clusters with *a priori* information about the "real" communities. However, as stated before, this information is often only partially available and depends on the subjective appreciation of individuals. A more relevant alternative is to design an automatic procedure to probe the relevance of the clusters identified by a given algorithm, using only the information provided by the network and the clustering algorithm

Such a procedure should in particular answer these three crucial questions in order to evaluate the validity of the community structure of a network.

1. Which are the nodes lying at the interface between different clusters (so called "unstable" nodes)?

2. Is there a global measure of the sensitivity of the community structure to small changes in the network?

3. Is the community structure more relevant than the one found in a random network?

In this Chapter, we first describe in details our method to answer the three questions above, illustrating it with several examples. Then we apply it to a network built from the relation of synonymy between words and we show how unstable nodes can often be interpreted as ambiguous words. Another example of the central role of unstable nodes in order to uncover the structural properties of a network is presented in Chapter 5 for the di-alanine network.

4.2 Adding noise to a network

In order to analyze the stability of a system a very common approach is to introduce a small perturbation and to see how the system reacts to it. This approach is especially appropriate for dynamical systems described by an energy function H and whose solutions are given by the extrema of H. However in the case of a clustering algorithm, there is often no way to define a function whose minima describe the clusters (except for the Potts clustering [149] and the modularity optimization [124], see Chapter 3) and no analytical approach can be carried out to probe the stability of a solution. Hence experimental perturbations are the most natural way to investigate how robust the results are. The central idea behind an experimental approach is to slightly modify either the algorithm itself or the network and to compare the different results obtained after each perturbation. A first attempt was proposed by Tyler, Wilkinson and Hubermann [170, 182] modifying slightly the Girvan-Newman algorithm [66]. The Girvan-Newman algorithm is based on the successive removal of edges with high edge betweenness. When two edges have exactly the same betweenness, it is not clear which one should be removed. Exploiting this ambiguity, Wilkinson and Hubermann have compared the community structure resulting from different choices for the removal of edges with equal betweenness. More recently several non-deterministic clustering algorithms have been developed [149, 75, 44, 191]. Using the different possible outputs, it is possible to compare the different runs of the algorithm and to see whether the results are stable or not. However all those approaches strongly depend on a particular clustering algorithm and do not allow to tune the amount of noise added in the process.

To fill this gap, we have introduced a well-defined method to evaluate the stability of the community structure that does not depend a particular clustering algorithm [61]. The idea is to add a random noise over the edge weight in the network and to compare the clusters arising from different noisy realizations of the network. We stress that noise is not only a useful tool to probe the cluster stability, but has actually a deeper interpretation in the context of complex networks. In many real-world networks, edges are often provided with some intrinsic weight, but usually no information is given about the uncertainties over these values. Adding some noise could fill this lack, although arbitrarily, to take into account the possible effects of uncertainties.

Two ways of adding noise have been designed. Initially, the noise was distributed uniformly over the entire network. If w_{ij} is the weight of the edge between node i and node j, the amount of noise added to it is randomly selected between $-\sigma \cdot w_{ij}$ and $\sigma \cdot w_{ij}$, where σ is a fixed parameter, $0 < \sigma < 1$. Although this way of adding noise performs well, it was noticed later that sometimes results could be improved by taking into account the heterogeneity in the degree distribution. In this modified version of the algorithm noise is added on each edge with weight w_{ij} as $+\sigma_{ij}$ with probability $\frac{1}{2}$ and $-\sigma_{ij}$ with probability $\frac{1}{2}$, where $\sigma_{ij} = w_{ij}\left(1 - \frac{1}{\sqrt{\min(k_i,k_j)}}\right)$. The 'min' function is used to allow for large σ_{ij} only if both nodes i and j have a large degree.

All applications presented in this Chapter are based on the homogeneous way to introduce noise. In Chapter 5 an example of the heterogeneous noise is discussed in details. Although the two ways of introducing noise differ slightly, the underlying idea of perturbing a network is indeed the same in both cases.

4.3 Unstable nodes

In order to understand how the cluster structure changes with noise,we first introduce the *"in-cluster probability"* p_{ij}. p_{ij} is defined as the fraction of times two neighbor nodes i and j have been classified in the same cluster during several occurrences of the clustering algorithm on different noisy realizations of the network. For example in Figure 4.1, node 7 has been classified 51% of the time in the same cluster as node 6 and 49% of the time in the same cluster as node

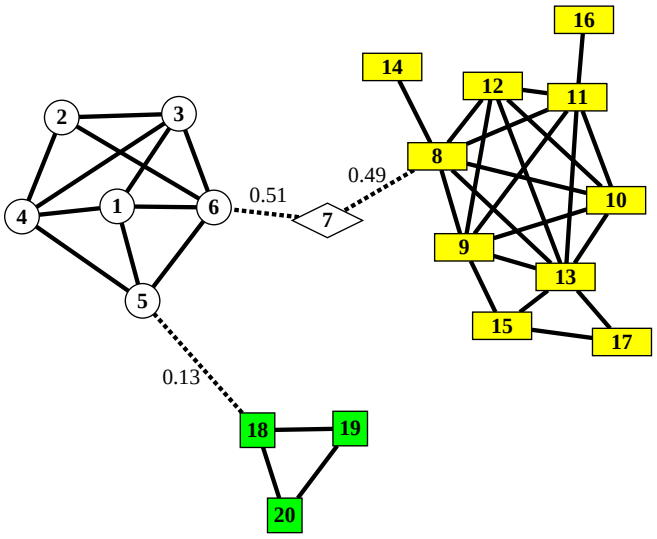


Figure 4.1: Small toy network with one unstable node (7). The clusters obtained without noise are labeled with different colors. Only $p_{ij} < 0.8$ are shown (dashed edges). We have used MCL with $r = 1.6$, and $\sigma = 0.5$

8. Hence $p_{67} = 0.51$ and $p_{78} = 0.49$. Edges with p_{ij} equal to one are always within a cluster and edges with p_{ij} close to zero connect different clusters.

At this point the first question about unstable nodes can be addressed. Intuitively these nodes are expected to be surrounded by edges with p_{ij} close to 0.5. In order to implement efficiently this idea, we established the following algorithmic procedure.

First edges with p_{ij} lower than θ are defined as "*external edges*" (typically the threshold θ is chosen as $\theta = 0.8$). Removing external edges of the network yields a new, most often disconnected, network. We call *initial clusters* the clusters obtained without noise, and *subcomponents* the disconnected parts of the network after the removal of external edges. If the community structure is stable under several repetitions of the clustering with noise, subcomponents correspond to initial clusters. In the opposite case, a new community structure appears with some similarities with the initial one (see Figure 4.1). A crucial step is to find which subcomponents correspond to the initial clusters and which consist of unstable nodes. If E_1 (resp. E_2) is the set of initial clusters (resp. the set of subcomponents), we define the similarity (s_{ij}) between the initial cluster $C_{1j} \in E_1$ and the subcomponent $C_{2i} \in E_2$ as the Jaccard index:

$$s_{ij} = \frac{|C_{2i} \cap C_{1j}|}{|C_{2i} \cup C_{1j}|},\ 1 \leq i \leq |E_2|,\ 1 \leq j \leq |E_1|.$$

For instance $C_{1j} = C_{2i}$ implies that $s_{ij} = 1$ while $C_{1j} \cap C_{2i} = \emptyset$ yields $s_{ij} = 0$. Using s_{ij}, the identification of subcomponents corresponding to initial cluster can now be performed. To each initial cluster $C_{1j} \in E_1$ we associate the subcomponent $C_{2i}, 1 \leq i \leq |E_2|$ with the maximal similarity (most often C_{2i} corresponds to the stable core of the initial cluster C_{1j}). If there are more than one subcomponents, none of them will be associated with the initial cluster. In practice, this case is extremely rare.

To exemplify the procedure, let us apply it to the network in Figure 4.1. For this network, three initial clusters (the three colors) have been identify by MCL without noise. After 100 runs of MCL with a noise given by $\sigma = 0.5$, p_{ij} values have been computed as shown in Figure 4.1 and four subcomponents ({1,2,3,4,5,6}, {7}, {8,9,10,11,12,13,14,15,16,17}, {18,19,20}) appear after removing edges with $p_{ij} < \theta = 0.8$. The comparison based on the similarity measure

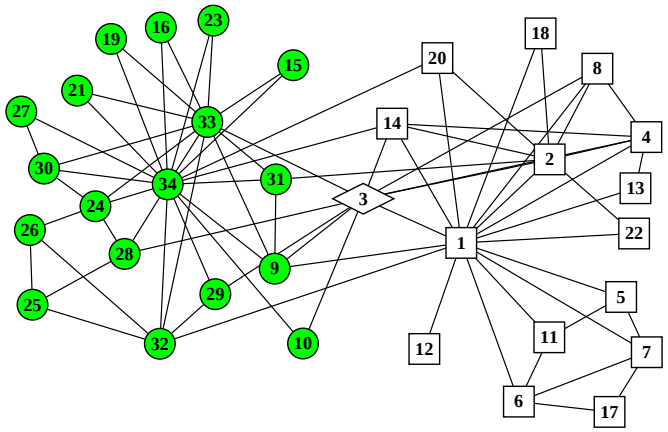


Figure 4.2: Zachary's karate club network. The two clusters are represented with two different colors. The unstable node is represented by a diamond. We have used MCL with $r = 1.8$, and $\sigma = 0.5$

associates the three biggest subcomponents with the three initial clusters, while subcomponent $\{7\}$ is not associated with any cluster, as expected from a visual observation.

Nodes belonging to subcomponents that have never been associated with any initial cluster are defined as *unstable* nodes. In the previous example only one node is unstable but it may happen as well that unstable subcomponents consist of more than one node, as in Figure 4.6.

To further illustrate the method for detecting unstable nodes, we studied the "karate club network" (Figure 4.2), which has often been used as a benchmark network for several kinds of clustering algorithm. In this network, nodes represent the members a karate club [189]. At a certain moment the club split due to internal disagreement. Edges represent social interactions some time after the split. MCL correctly identifies the two communities, which correspond to the actual division of the club. The only unstable node is represented with a diamond. This node is connected to four nodes of one community and five of the other one. It is therefore absolutely justified to consider it as an unstable node since it corresponds to an individual who still had contact with people from the two groups.

4.4 Clustering entropy

Our algorithm to identify unstable nodes allows to answer the first question about nodes lying between clusters. The two following questions concerned the validity of the community structure identified by a clustering algorithm. Locally those questions can be addressed by looking at the in-cluster probabilities of the edges inside each cluster and around a cluster. For instance if all edges inside a cluster have p_{ij} close to one and all edges connecting a cluster to its neighbors have p_{ij} close to zero, the cluster is rather stable. However for large networks such a local evaluation can be extremely lengthy and not easy to use in practice. To have a more global measure, we introduce the *Clustering Entropy* of edges, defined as:

$$S = \frac{-1}{M} \sum_{(i,j)} \{p_{ij} \log_2 p_{ij} + (1 - p_{ij}) \log_2 (1 - p_{ij})\}, \tag{4.1}$$

where the sum is taken over all edges and M is the total number of edges in the network [61]. If the network is totally unstable (i.e. in the most extreme case $p_{ij} = \frac{1}{2}$ for all edges), $S = 1$, while if the edges are perfectly stable under noise ($p_{ij} = 0$ or 1), $S = 0$.

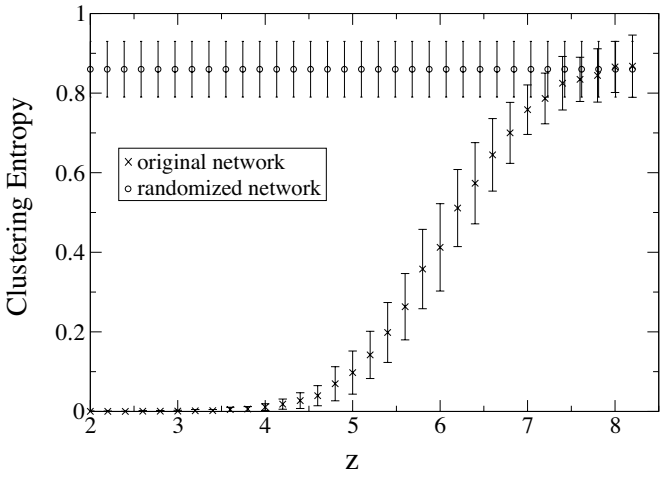


Figure 4.3: Clustering entropy as a function of z, for a benchmark network with 4 communities of 32 nodes (see main text). Error bars represent the standard deviation for different networks with the same z. MCL has been used with $r = 1.85$, and $\sigma = 0.5$

The choice of the name "entropy" refers to the analogy with a 2-state spin system on the edges of the network, with p_{ij} the probability of having spin $+1$. Assuming in first approximation that the p_{ij} are independent of each other (mean-field approximation), the sum in Eq. (4.1) is equivalent to the sum over all possible configurations of $p \log p$, p being the configuration probability.

The clustering entropy is thus a global measure of the sensitivity of the community structure to an external noise characterized by the parameter σ.

In addition the clustering entropy allows for comparing with a network without predefined cluster structure. To avoid biasing the comparison, the clustering entropy of a network is always compared with the one of a randomized version of the network in which the degree of each node is conserved [106, 49]. The randomized network plays the role of a null-model since clusters (if present) are destroyed by the randomizing process. Note however that we do not assume the randomized network to have no apparent community structure [76, 150], we only quantify the difference between the clustering entropy of the original network and the randomized one. If this difference is important, it shows that the network has an internal cluster structure that differs fundamentally in terms of stability from a network where nodes have been connected randomly.

To illustrate the principle of the comparison based on the clustering entropy, we applied it on the well-known benchmark network introduced first in [66]. The network consists of 4 communities of 32 nodes each. The nodes are connected with a probability P_{in} if they belong to the same community and P_{out} if not. Typically P_{in} and P_{out} are changed keeping the average degree of the nodes to a constant value of 16. Figure 4.3 shows the clustering entropy of the network. Parameter z is the average number of edges connecting a node from a given cluster to nodes of other clusters ($z = 96 \cdot P_{\text{out}}$). For small z clusters are very well defined and most algorithms correctly identify them. As z increases, clusters become more and more fuzzy and for $z > 7$ even the best currently available algorithms fail to recover the exact cluster structure of the network (actually the cluster structure tends to disappear from the network). This corresponds to the point from which the comparison of the clustering entropy does not allow to differentiate between the network and a randomized one. Error bars in Figure 4.3 stand for the standard deviation and give an indication of the dispersion of the values for different realizations of the network.

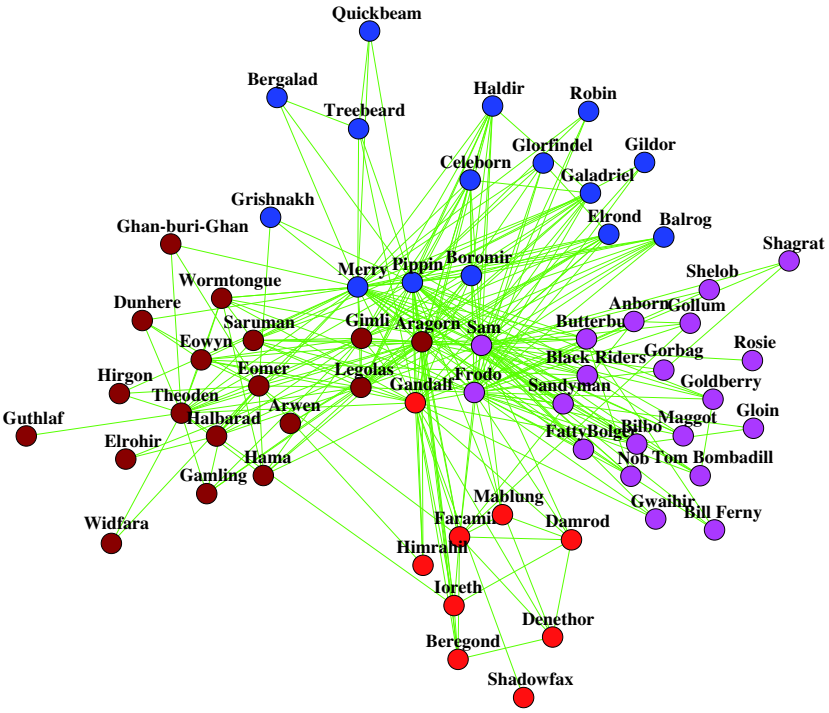


Figure 4.4: Community structure of the network composed of the main characters appearing in the *Lord of the Rings*. The four different colors are the four clusters identified by optimizing the modularity, as described in [33].

An interesting counter-example is provided by a network constructed from the book of Tolkien, the *Lord of the Rings*. The nodes of the network are the main characters of the book. Two nodes are connected if the characters (human, hobbits, elves,...) interact significantly with each other along the story. By interacting significantly, we mean that they perform some task together or speak together. In Figure 4.4 the nine members of the Fellowship of the Ring have been placed in the middle of the network, since they play a central role in the story. After having built the network, we applied the clustering algorithm based on the greedy optimization of modularity [33]. The results are shown in Figure 4.4 and, quite impressively, the four clusters identified with this algorithm reflect exactly the split in the Fellowship that occurs at a certain time in the story. However a closer inspection of Figure 4.4 seems to indicate that the clusters are not obvious, at least for the nine central nodes. More insight is gained by comparing the clustering entropy with the one of a randomized network. Setting the amount of noise to $\sigma = 0.5$ yields to $S = 0.61$ and $S_{\text{rand}} = 0.77$. Both values are rather large and close to each other, which indicates that the community structure is extremely unstable and therefore should not be trusted. Thus the only possible conclusion is that the fellowship of the ring was indeed unstable!

4.5 Synonymy networks

Most of the work about the robustness of the clustering was initiated by the study of a special kind of networks based on the synonymy relation between words [62]. The notions of synonymy and polysemy of words are common to every reader. However it induces several ambiguities in automated text mining technique. The aim of Word Sense Disambiguation(WSD) is precisely to associate a specific sense to every word within a context [160] and for each word in each context to have a list of possible synonyms. Our starting point to build a network of synonyms was a

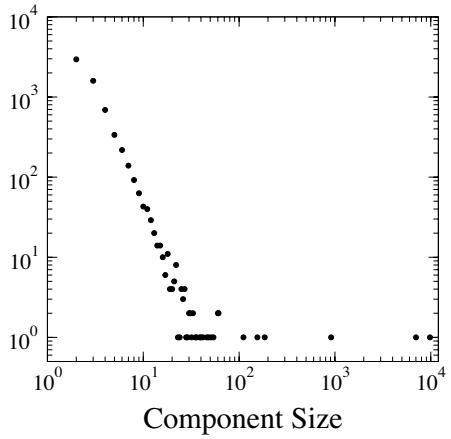


Figure 4.5: Distribution of the component size in the synonymy network.

dictionary of synonyms. Yet following naively the synonymy relation throughout the dictionary induces several problems. In particular if word W_1 is a synonym of W_2, W_2 a synonym of W_3, ... and W_{N-1} a synonyms of W_N, it is likely that W_1 is no longer a synonym of W_N for large N. Thus the synonymy relation is not transitive. We will see that the use of network representation helps to overcome this problem.

Building the network

The network under consideration has been built from a French dictionary of synonyms by Jean-Pascal Pfister [144]. The synonymy relation is defined between words in one of their senses and is considered to be symmetric. The resulting network is thus undirected and unweighted. It is not fully connected but consists of many disconnected components, whose distribution is displayed in Figure 4.5. If the relation of synonymy was perfectly transitive, one would find within each component that all nodes are synonyms.

A human evaluation shows that the small components of the network are made of words whose sense is very close. However some components are made of almost 10'000 words (see Figure 4.5) that can not be considered as synonyms, even in a very weak sense. For instance the word *fêtard* ("merrymaker") and *sans-abri* ("homeless") (see Figure 4.6) appears in the same component. Thus even if a path exists between two nodes, the slight changes in the senses that may occur at each step along this path often result in a quite different sense between both ends. Hence the component structure does not allow to recover the correct classes of synonyms. Nevertheless the large components clearly show sub-structures as in Figure 4.6, suggesting that a partition into smaller communities might give more reasonable results.

4.5.1 Community structure and robustness analysis

To uncover the internal organization of the synonymy network, MCL has been applied to the largest components using $r = 1.6$. As an example, the community structure of a component of size 185 is shown using different colors in Figure 4.6. The obtained subdivision into smaller clusters is definitely more meaningful; e.g. *fêtard* is no longer in the same cluster as *sans-abri*.

After applying MCL the size of the largest clusters is much smaller than it was for the components and a human evaluation shows that within the clusters, words can be considered most often as real synonyms.

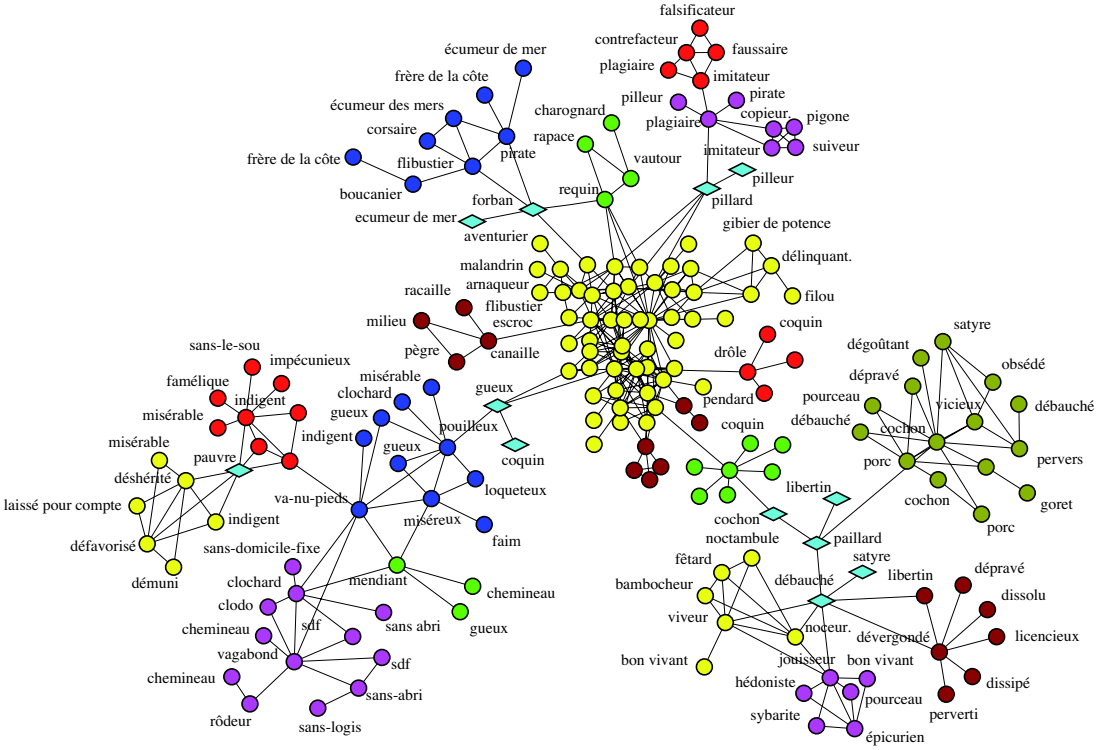


Figure 4.6: Community structure of a component with 185 nodes from the synonymy network. The different colors represent different clusters obtained by MCL with $r = 1.6$. Unstable nodes are represented with diamonds. The same word may appear more than once since different senses were sometimes already present in the dictionary.

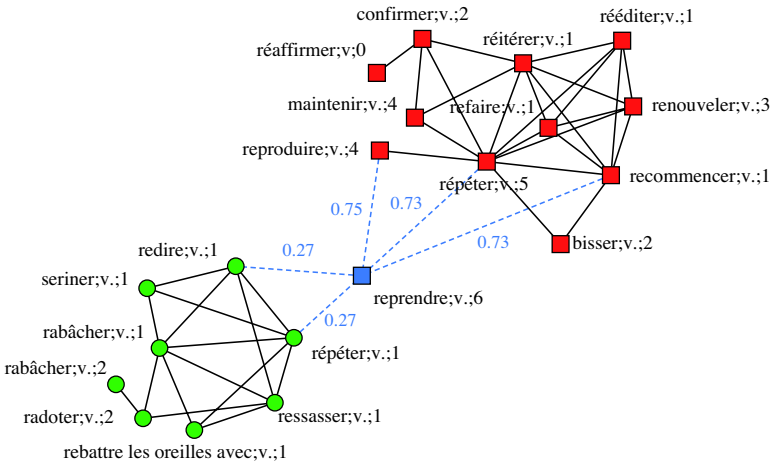


Figure 4.7: Small sub-network with one unstable node ("*reprendre*"), extracted from a component of 111 nodes. The values over the dashed edges are the probabilities for the edges to be inside a cluster (average over 100 realizations of the clustering with $r = 1.6$, and $\sigma = 0.5$.). Only probabilities smaller than $\theta = 0.8$ are displayed. The shape of the nodes indicates the clusters found without noise. The symbols $v.$ stands for verbs and numbers after the word indicate which sense has been considered.

Ambiguous words and unstable nodes

After partitioning the network into clusters, we performed an analysis of the robustness of the community structure.

The first objective was to probe how reliable the community structure is from a topological point of view. In the framework of a synonymy network, unstable nodes, may correspond to polysemic words which have not been clearly identified as such (i.e. one of their senses is not present in the dictionary). 100 runs MCL have been performed over the largest components of the network each time with an amount of noise given by $\sigma = 0.5$. An example of the results is shown in Figure 4.7. The two main clusters appear clearly. Green nodes refer to "saying again something", while most red nodes signify "doing again something". The blue node in the middle has both meanings in French, and therefore should be assigned to both groups.

To refine the synonymy database under study, unstable nodes have been split among their adjacent subcomponents. The adjacent subcomponents are defined as the subcomponents to which the node is connected through at least one edge with a probability higher than a given threshold θ'. Typically $\theta' = 1 - \theta$, where θ was the threshold for defining an edge as external. If several unstable nodes are connected together, we split them according to the following procedure: first group these nodes keeping only the edges with $p_{ij} > \theta'$; then, for each group, duplicate it and join it to its adjacent subcomponents (see Figure 4.8).

As a validation of our results about unstable nodes, we computed the clustering coefficient [181] and the betweenness centrality [119] of these nodes. Averaging over the whole network, a clustering coefficient of 0.26 has been found for the unstable nodes and 0.45 for the other nodes. The betweenness centrality of unstable nodes is on average 1.6 times larger. This important difference was expected since unstable nodes often lie between clusters, and therefore usually do not have a large clustering coefficient, but have a large betweenness (see Figure 4.7 and 4.8). To further illustrate the relation between p_{ij} and betweenness centrality we plot the edge betweenness versus p_{ij}. Figure 4.9 shows that edges with a low p_{ij} have on average a larger betweenness, which is consistent with the Girvan-Newman clustering algorithm [66].

Finally a comparison between the clustering entropy of the network with the one of a ran-

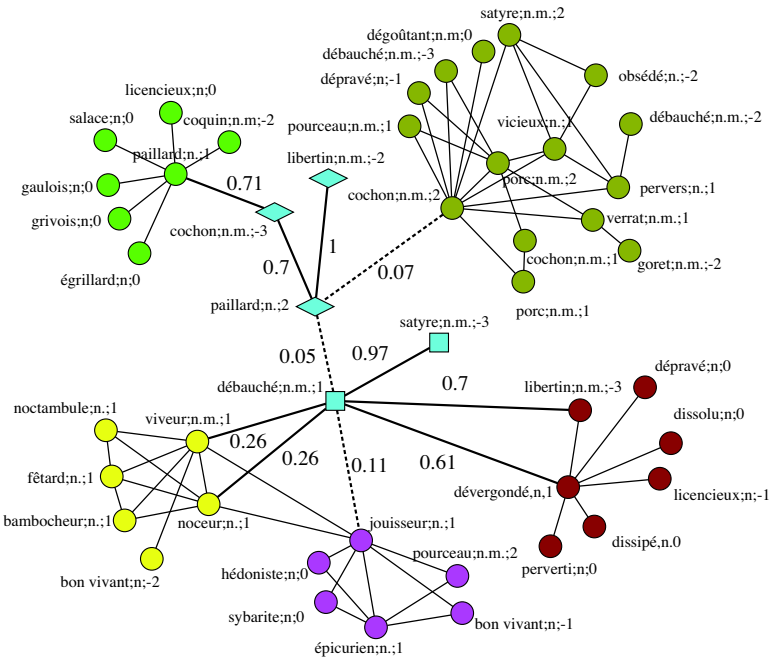


Figure 4.8: Zoom over the bottom-right of Figure 4.6. Five unstable nodes have been found (cyan squares and diamonds). To split them among their adjacent clusters, we proceed as follows: we first removed the edges with $p_{ij} < 1 - \theta$ (dashed-line), which produced two groups of unstable nodes ({*cochon;-3 libertin;-2, paillard;2*} (diamonds) and {*débauché;1, satyre;-3*} (squares). The first group has only one adjacent subcomponent. It is therefore merged to this subcomponent. The second group has two adjacent subcomponents. It is thus duplicated and merged into those two subcomponents.

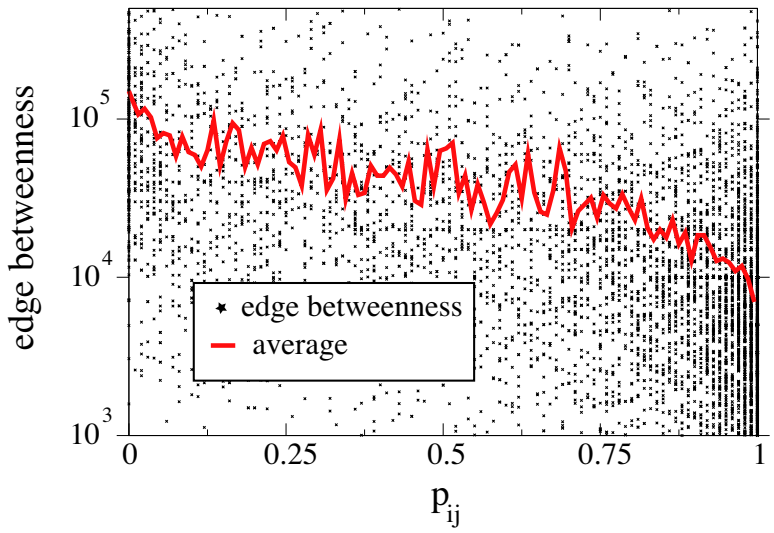


Figure 4.9: Edge betweenness versus p_{ij} for a component of 9997 nodes from the synonymy network, $\sigma = 0.5$.

Component Size	S	S_{rand}
912	0.25±0.01	0.55±0.01
185	0.19±0.01	0.62±0.02
155	0.27±0.01	0.55±0.03
111	0.21±0.01	0.69±0.02
61	0.20±0.01	0.68±0.04
60	0.19±0.01	0.76±0.04
54	0.21±0.01	0.60±0.07
51	0.21±0.01	0.69±0.05
average	0.21	0.64

Table 4.1: Comparison for several components. S is the clustering entropy of the original components. S_{rand} is the average clustering entropy for 50 randomized versions of the components. $\sigma = 0.5$.

domized network has been performed. The randomized network is obtained by reshuffling edges in way that the degree of each node is preserved (see [106]). Table 4.1 shows the comparison for several big components of the network of synonyms. The clustering entropy of the randomly rewired components is at least twice bigger than the clustering entropy of the original components. From the results in Table 4.1, we can conclude that the clusters obtained with MCL are not an artifact of the algorithm, but correspond to a real community structure of the network.

4.6 Discussion

In its initial version (the one that was applied in this Chapter), the method to insert noise into the network depends on the parameter σ. This parameter tunes the amount of noise added in the network. With σ close to zero, unstable nodes can not be detected, while with σ close to one, the topology of the network changes dramatically. For a significant perturbation of the network or if the edge weight has a large intrinsic uncertainty, a rather high σ is indicated, while for small perturbations a low σ is more appropriate. The possibility of tuning σ can be an advantage in order to evaluate the effect of different amount of noise in a network. However in some situation one is interested in finding exactly the unstable nodes and the clustering entropy of a network. Instead of fixing arbitrarily the value of σ, our second method to introduce a heterogeneous noise depending on the degree of the nodes at each end of an edge would be more appropriate. An example is discussed in details in Chapter 5, in which unstable nodes could be double-checked using *a priori* information about the network.

The parameter θ plays a role for identifying the unstable nodes. It has to be interpreted as a threshold such that two adjacent nodes that have been classified in the same cluster with p_{ij} larger than θ can be considered as belonging to the same cluster. The large θ, the higher the confidence about clusters. For the synonymy network, the choice of $\theta = 0.8$ was motivated by the following reason. θ should not be too close to one to avoid insignificant effects due to a peculiar noisy realization of the network, neither too close to 0.5, since if θ equals 0.5 subcomponents basically correspond to initial clusters (see Figure 4.1).

Finally we stress that the time consuming step is the computation of p_{ij} involving only the parameter σ, since we have to repeat several times the clustering. Therefore probing different values of θ can be done without decreasing the speed of the algorithm.

4.7 Conclusion

In this Chapter we have addressed the question of evaluating the reliability of the community structure. As clustering techniques become more and more used to organize and reduce the complexity of large data sets, this question is indeed crucial. Moreover the absence of completely satisfactory definition of clusters often hampers a direct validation of clustering algorithms output. Thus more involved approaches had to be designed. We have shown in this Chapter that the introduction of noise over the edges and the definition of p_{ij} provides a well-defined and objective way to identify unstable nodes and to distinguish between a true modular structure and artifacts of clustering algorithms. Finally the clustering entropy allows for a quantitative comparison between a network and a null-model.

In contrast with most existing methods to evaluate the robustness of the community structure in complex networks, the use of external noise into the network is very general and does not depend on a particular algorithm. In this sense it can be applied with any existing method and therefore represents a step forward to a better understanding of the reliability of clustering algorithms.

Finally from a computational point of view, the method requires to run several times the clustering algorithm. Although this can be time consuming, it is straightforward to parallelize in order to apply it to very large networks.

Chapter 5

Configuration Space Networks

5.1 Introduction to Configuration Space Networks

Experimental studies of complex systems, such as high-throughput experiments in biology, web crawls, etc. are the most common information sources to construct complex networks. For this reason systems naturally represented as networks are composed of a large number of units interacting between each other. Interactions depend on the system under scrutiny, but often are rather straightforward to define (interactions between proteins, hyper-links between web pages,...). However there exists a large class of complex systems, in particular dynamical systems, for which the complexity arises not through the large number of system units, but rather from the complicated spatio-temporal system behavior. In structural biology for instance, even a simple peptide exhibits a complex dynamics involving a large number of degrees of freedom. Complementary to experimental techniques (mainly X-ray diffraction or NMR) it has been shown already more than 30 years ago that lots of insights into the system can be obtained by exhaustive numerical simulations [98]. In particular, the folding of proteins has been described as a stochastic process whose dynamics is governed by a free-energy landscape [70, 5, 56, 167], which offers the possibility to apply various tools of statistical physics to study protein dynamics. Computer simulations, often refereed to as Molecular Dynamics (MD) simulations, have by now become widely used to analyze the dynamics of peptides or proteins and results have been successfully confronted with experiments [37, 174]. The advantages of simulating a system are countless and have opened new ways to verify, understand and predict the system behavior. Yet, simulations do not solve all issues, and sometimes the simulation results form themselves a highly complex system. This is especially true in MD simulations exploring complex and multi-dimensional free-energy landscapes consisting in different energy basins. The dimension of the phase space increases dramatically with the peptide size and following the dynamics is like following a path in a high dimensional space, with hundreds of re-crossings, turn-overs,... In addition the high dimension becomes an important hurdle for valuable visualization of the process. A common way to avoid the problem of high dimension is to project the free-energy landscape onto one or two order parameters. Unfortunately the choice of order paramteres often can not be done unambiguously and certainly looses some information about the system [138, 19, 101, 16, 92, 35, 26].

To tackle this high complexity, new approaches based on graph theory and complex networks have been developed recently. Krivov and Karplus have introduced a method based on disconnectivity graphs (DG) for the analysis of unprojected free-energy surfaces of short peptides using an equilibrium MD trajectory [90, 91, 92]. They have developed the DG approach by relating the total rate between two free-energy minima (considering all possible pathways) to the maximum flow through a network with weighted edges. This technique has been applied

to the configuration space of a tetrapetide and the folding of a simple hairpin of protein G. At the same time, energy landscapes have been represented as complex networks. Doye has applied graph analysis for the study of the potential energy minima of a Lennard-Jones cluster of atoms [42]. In his work the minima of energy landscape are the nodes of a network and saddle points between two minima are represented by edges. The topological features of such networks have been investigated in details in [107, 108].

Graph theory has also been applied to study the dynamics of a domain of G proteins [4]. Each configuration observed in the simulation is mapped onto a node and nodes are connected according to their structural similarity. Andrec *et al.* [4] suggested that random walks on this kind of networks reflect the peptide dynamics. More recently a set of helical proteins [83] have been investigated with the tools of graph theory. The native state of each protein forms a node of a network, and connected components of the graph reveals the groups of proteins with a similar native state. Previously, the configurations of a lattice polymer had also been mapped onto a network [158]. However in these three last studies, no consideration was given to the dynamics and edges were placed between configurations that are structurally close to each other.

The Configuration Space Network (CSN) [148] is a recent technique to analyze and visualize complex high-dimensional simulations. It applies typically to the simulations of peptides or proteins, but can be extended to any dynamical system. Assuming that the ensemble of configurations is discrete or at least that the configuration space can be partitioned into discrete configurations (see further for a technical discussion about this partition), the result of a simulation is a chronological time series of the configurations visited along the simulation. The main idea of CSNs is to consider each configuration as a nodes of a network. With the aim of following as closely as possible the dynamics, an edge between two nodes is drawn if a direct transition between the two configurations was observed along the simulation. Finally the edge weight is set equal to the total number of transitions between the two configurations, and thus is proportional to the transition probability. Therefore, by construction of CSNs, random walks on such networks describe the peptide dynamics, as observed in the simulation.

The aim of this Chapter is to understand the relation between the topology of CSN and the architecture of the free-energy landscape [64]. Two essential aspects of CSN have been addressed. In Section 5.2 the community structure is discussed in details. Because of their generality to describe any simulation, CSNs are first illustrated for Monte Carlo (MC) explorations of simple energy landscapes. Then more complex and more realistic examples are addressed in Section 5.3. The second important aspect, though not immediately related with the scope of this Thesis, is the topology of CSNs. In Appendix A a complete discussion about the weight distribution in CSNs is presented [64]. Other features such as degree distribution, clustering coefficient or degree correlation have been recently described in [65].

5.2 Community Structure of CSN

One of the most crucial characteristics of energy landscapes is the presence of energy basins. Energy basins dramatically affect the dynamics, since the simulated protein spends most of its time within basins and performs only a few transitions between different basins. In the terminology of CSNs, nodes representing configurations lying in the same energy basin have several connections between each other, and much less to nodes in other basins. The previous statement implies that basins should appear as communities of CSNs. To validate this prediction, simple free-energy landscapes explored with Monte Carlo simulations are first studied, with an emphasis on their community structure. Although this constitutes an over-simplification compared to MD simulations, the general principles underlying the dynamics are the same. Additionaly the use of free-energy landscape models provides crucial benchmarks to test any new method.

5.2.1 CSN from Monte Carlo simulation

Monte Carlo (MC) techniques are the core of all discrete simulation processes. Stochastic processes, in particular random walks, on a D-dimensional energy landscape $U(\mathbf{x})$ are often simulated using MC methods. To achieve this goal the energy landscape is first discretized into a hyper-cubic lattice with a distance a between two neighbor sites. The evolution of the simulation proceeds as follows: at each time step a neighbor site on the lattice is chosen randomly and the move is accepted with a probability given by the Metropolis rule [111]

$$p_{i\to j} = \min\left(1, \exp\left(-\Delta U_{ij}\right)\right)$$

where $\Delta U_{ij} = U(\mathbf{x}_j) - U(\mathbf{x}_i)$ and $U(\mathbf{x})$ is expressed in the units of $k_B T$. The metropolis rule ensures that detailed balance is preserved.

Building the CSN

A trajectory sampled by MC simulation consists of a chronological sequence of the sites visited during the dynamics. This chronological sequence describes the dynamics at a microscopic level since only nearest neighbor sites can be reached at each time step. In order to build a Configuration Space Network, snapshots are saved every M steps to form the time series. The sites of the lattice are the system configurations and correspond to the nodes of the network. Two nodes are connected if the simulation evolved from one site to the other one within M steps. When M approaches one only configurations spatially close to each other are connected, while for large M the timeserie becomes uncorrelated and the free-energy landscape exploration is equivalent to a random sampling with probability density $e^{-U(\mathbf{x})}$. In other words, a link is a temporal relation between configurations at a time-scale given by the parameter M. Values of M between 10 and 50 allow to relax the microscopical constraints due to the lattice, while still preserving the interesting correlations along the time series (see [65] for a detailed discussion about the parameter M).

CSN networks are weighted networks since an edge may be visited more than once. The weight of an edge $w_{i\to j}$ is defined as the number of direct transitions from node i to node j. Similarly the weight of nodes accounts for the number of times the configuration has been visited.
Both the presence of an edge and its weight are essential to characterize the network topology and dynamics. In particular the quantity $w_{i\to j}\left(\sum_l w_{i\to l}\right)^{-1}$ is equal to the probability of moving from i to j. For this reason random walks on the CSN describe the same process as the MC (or any other kind of simulations) exploration of the energy landscape $U(\mathbf{x})$. Hence complex energy landscapes can be studied by considering only the corresponding CSN.

Multi-well energy landscape

The first model of energy landscape consists in several basins. It is given by the multidimensional double-well shown in Figure 5.1,

$$U(\mathbf{x}) = 5\sum_{i=1}^{D}(x_i^4 - 2x_i^2 - \epsilon x_i + 1) \tag{5.1}$$

This landscape is characterized by 2^D minima, where D is the dimension of the system. The parameter ϵ introduces an asymmetry between the minima, such that some minima have a lower energy than others.

Given $U(\mathbf{x})$, the system dynamics is simulated using the Monte Carlo protocol described earlier on a D-dimensional lattice. Snapshots are saved every $M = 5{\cdot}D$ steps. The CSN obtained with D=3 is first shown in Figure 5.2. a has been set equal to 0.2 and $\epsilon = 0.05$. The resulting

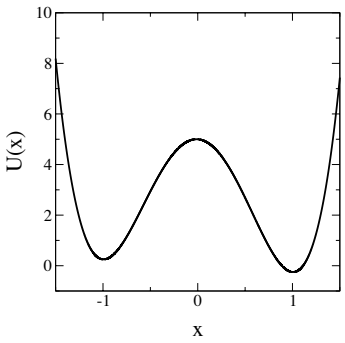


Figure 5.1: One dimensional view of the multiple well energy landscape of Eq. (5.1), $\epsilon = 0.05$.

network is made of 1752 nodes and is displayed in Figure 5.2 (node size is proportional to their weight). Remarkably the "cubic" shape of the network in Figure 5.2 reflects the internal organization of the energy basins. More quantitatively, MCL [172] finds 8 clusters represented with different colors. The 8 clusters correspond to the 8 energy basins of the 3D-potential with less than 5% of errors.

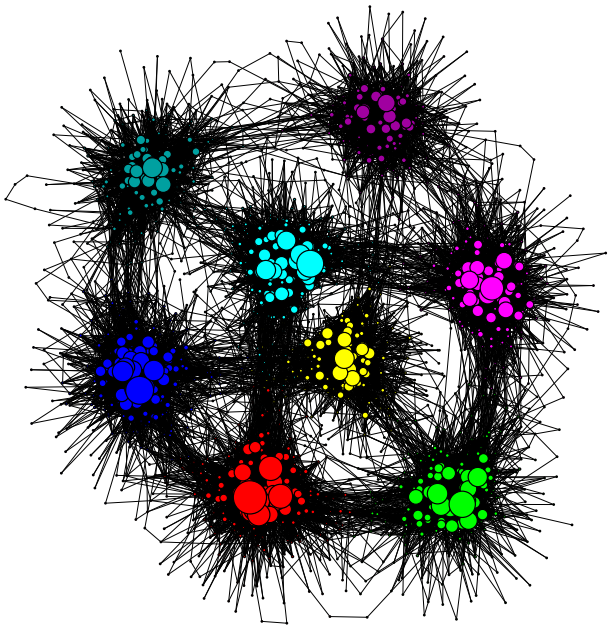


Figure 5.2: CSN obtained from a MC simulation along the potential of Eq. (5.1) in $D = 3$ dimensions. $a = 0.2$, $\epsilon = 0.05$, $N_s = 10^5$. Colors show the different clusters found with MCL ($r = 1.2$).

To further illustrate the power of CSNs when analyzing and visualizing high dimension energy landscapes, the same potential of Eq. (5.1) is taken in $D = 5$ dimensions this time. All parameters are the same, except $a = 0.3$ and the simulation length $N_s = 1.5 \cdot 10^6$ in order to improve the exploration. The network obtained from the MC simulation is made of 24'747 nodes and 777'853 links, which hinders almost completely a valuable visualization. Instead the network of clusters is displayed in Figure 5.3A. Each node corresponds to a cluster in the

initial network, and edges are drawn if an edge existed between two nodes of each clusters (for clarity self edges are not shown). Node size is proportional to the total weight of the clusters and edge size to the sum of the edge weights connecting the two clusters. Remarkably MCL finds 32 clusters, as expected from the 32 minima. Comparing the cluster to which nodes have been assigned and their position in the **x** space shows that only 3.3% of the nodes have been missclassified, i.e. they do not belong to the same cluster as the other nodes of their basin. In Figure 5.3A the color and the position of the nodes refer to their weight: clusters with similar weights are displayed with the same color. The cluster weight distribution is further illustrated in the histogram of Figure 5.3B. 6 groups of clusters are clearly identified. As expected, there is one heaviest node (red node) corresponding to the sites with only positive coordinates, five nodes of almost the same size (blue nodes) corresponding to sites with one negative coordinate, ten nodes (green nodes) to sites with two negative coordinates, ten nodes (pink nodes) to sites with three negative coordinates, five nodes to those with four negative coordinates (yellow nodes) and one last node (turquoise) to sites with only negative coordinates.

In this case the use of network representation is clear. From Figure 5.3A one can grasp the main features of the process, while several 2D-projections on the different axes would be needed to recover all energy basins and saddle points.

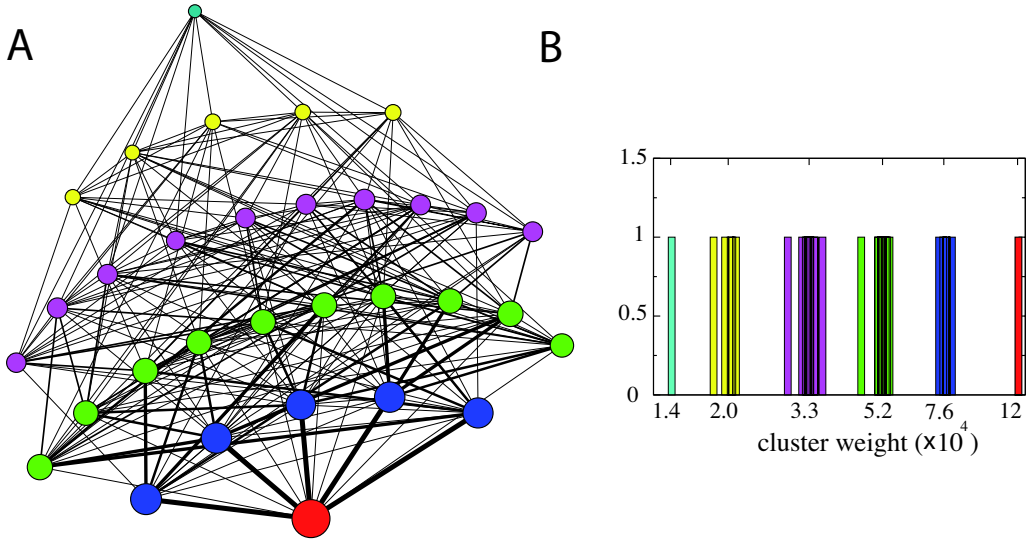


Figure 5.3: **A**: CSN of clusters obtained from a MC simulation along the potential of Eq. (5.1) in $D = 5$ dimensions. $a = 0.3$, $N_s = 1.5 \cdot 10^6$. Colors correspond to the different sizes of the clusters found with MCL ($r = 1.2$). **B**: Histogram of the cluster size (same color code).

Mexican Hat

In the previous example, energy basins are mainly *enthalpic*, i.e basins are characterized by a single minimum and have a funnel-like shape. Interestingly, a cluster analysis can detect as well the presence of *entropic* basins, i.e. regions in the energy surface without a single predominant attractor yet separated from the rest of the configurations of the system. An illustrative example is given by the Mexican-Hat landscape of Figure 5.4, which is defined in polar coordinates by the energy function

$$U(r) = 40(r^6 - 1.95r^4 + r^2) \tag{5.2}$$

In $D > 1$ dimensions, the model has two energy basins. One basin (the *central* basin) has

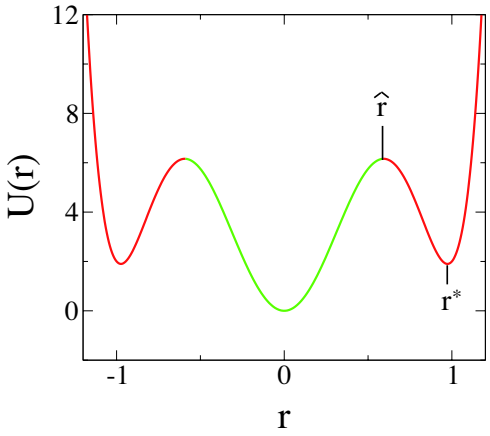


Figure 5.4: Plot of the Mexican Hat energy landscape of Eq. (5.2) along the radial coordinate r.

a minimum at $r = 0$. The other one (the *surrounding* basin) is a shell centered at $r = 0.97$. The two minima are intrinsically different. The central minimum is well defined and punctual, thus enthalpic. The second minimum corresponds to $r^* \approx 0.97$. It is not punctual and has an entropic part along Ω, where Ω is the solid angle in D dimensions, plus an enthalpic part along r. The two energy basins are separated by a maximum at $\hat{r} \approx 0.59$.

In Figure 5.5A the CSN of the mexican-hat is displayed. The network is obtained from the simulation on the landscape defined by Eq. (5.2) in $D = 2$ dimensions. The lattice was defined with $a = 0.02$ and snapshots were saved every $M = 10$ steps, resulting in $N = 1575$ different configurations visited by the simulation ($N_s = 10^5$). Two clusters have been found applying MCL with $r = 1.2$. From the picture the central cluster (green nodes) and the surrounding cluster (red nodes) are clearly visible and correctly reflect the two basins. In Figure 5.5 B the distribution of the radial coordinate r for each of the two clusters is shown. Less than 6% of the nodes are "misclassified", i.e. either belongs to the central cluster and have $r > 0.59$, or belongs to the surrounding cluster and have $r < 0.59$. Therefore the cluster structure of the network is consistent with the architecture of the energy landscape.

The two previous examples of CSN with simple energy landscapes show that indeed the community structure analysis is able to identify the different energy basins. In particular it is not restricted to basins with a single well-defined minimum, but performs as well if basins are characterized by a large number of lowest energy configurations (entropic basins).

Entropic basins are not easily identify and conventional techniques to find energy basin often fail. For instance steepest-descent algorithms always follow the out-going edge with the largest weight from any node. Disregarding the ambiguity arising when more than one edge have the same weight, this method performs rather well in the case of enthalpic basins. However steepest-descent algorithms are known to fail on entropic basins, mostly because of fluctuations from the equilibrium in the simulation. For instance some nodes lying at the minimum of the surrounding basin in the Mexican Hat example have a slightly larger weight than others, and steepest-descent algorithms often divide this basin into several smaller ones. Taking into account all edges, and not only the ones with the largest weights, clustering algorithms, and especially MCL which mimics a random walk on a network, allow to filter the irrelevant fluctuations inherent to any simulations and to retrieve the global features of the potential.

To summarize, CSNs and clustering algorithms have been shown to provide an interesting

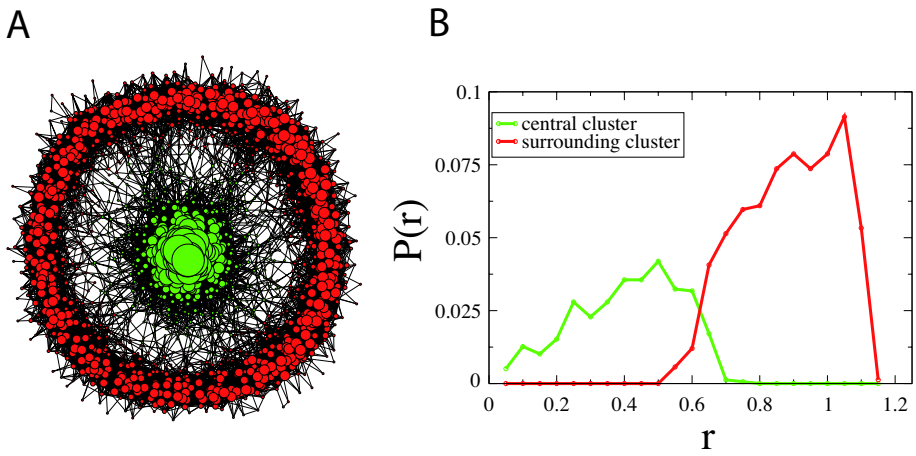


Figure 5.5: **A**: Network representation of the mexican-hat landscape model in $D = 2$ dimensions. Nodes are color coded according to the two clusters detected by MCL with $r = 1.2$, $a = 0.02$, $N_s = 10^5$. **B**: Distribution of the coordinate r of the nodes of the network for each of the two clusters.

alternative to study and visualize simple models of high-dimensional energy landscapes. In the next Section, we will validate this approach considering more complex systems.

5.3 The Di-alanine CSN

To extend the use of CSNs to more involved simulations, we invetigate the CSN built from MD simulations of a di-alanine peptide. The di-alanine peptide is a well-known benchmark system for evaluating new methods in the analysis of MD simulations [6, 19, 101].

5.3.1 MD Simulations

In the united atom representation the blocked alanine dipeptide is defined by 12 atoms (see Figure 5.6A). The main degrees of freedom are the dihedral angles ϕ and ψ of its two rotatable bonds. In the continuum solvent approximation used here four energy basins are found, designed as: C_{7eq}, α_R, C_{7ax} and α_L [6]. Those basins are shown in the (ϕ, ψ) free-energy landscape projection of Figure 5.6B.

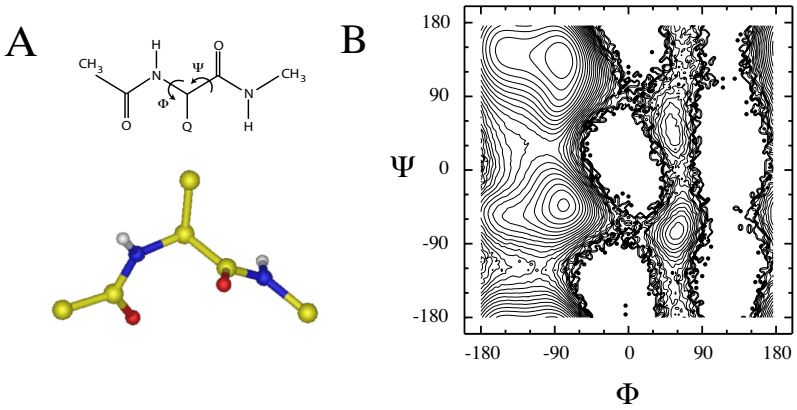


Figure 5.6: **A**: Chemical structure of di-alanine peptide (from the left to the right: $CH_3CONHCH(CH_3)CONHCH_3$) and united atom representation. **B**: Energy landscape of the di-alanine peptide projected on the two main degrees of freedom, as known from previous studies [6].

Five Langevin dynamics simulations with friction coefficient of 0.15 ps^{-1} of the alanine dipeptide were performed at 300 K for a total of 1 μs of simulation time. The integration time step was set to 2fs. Every trajectory was started from an extended configuration of the dipeptide. MD simulations were performed with the program CHARMM (PARAM19 force field) [25]. The electrostatic solvation free energy is calculated using an analytical approximation to the solution of the Poisson equation called ACE2 [159]. Non-polar contributions are approximated by a term proportional to the solvent accessible surface. Parameters were set according to the CHARMM implementation defaults. All MD simulations have been performed by Francesco Rao and Amedeo Caflisch at the University of Zürich.

5.3.2 Building the network

The time series was built from the simulation saving snapshots every $M = 10$ micro-steps. To define nodes and links of the CSN a discretization of the configuration space into small cells is needed. In this way, every snapshot sampled during the simulation is assigned to a cell of the discretized configuration space. Cells are the nodes of the network and direct transitions between them are edges. Several discretization approaches can be used to define the nodes

of a CSN from an MD simulation. For the alanine dipeptide the most natural discretization consists in partitioning the (ϕ, ψ) space (Ramachandran-map) into cells of equal size and label every snapshot visited during the simulation according to its (ϕ, ψ) value. A 50×50 division of the (ϕ, ψ) space gives a network of 1832 visited nodes and 54339 links (see Figure 5.7). Other discretization will be presented further in this section.

5.3.3 Community structure of the alanine CSN

The CSN of alanine displayed in Figure 5.7 provides qualitative insight on the architecture and dynamic connectivity of the landscape. It exhibits four densely connected regions which correspond to the free-energy basins of the dipeptide. Moreover, multiple pathways between basins emerge from the picture. C_{7eq} is connected to α_R by two independent pathways characterized by different populations where the statistically more (less) significant pathways corresponds to decreasing (increasing) values of ψ. There are also two independent pathways connecting C_{7ax} and α_L and two pathways between α_L and C_{7eq} one of which (via increasing ϕ) was observed only once in the five 200-ns simulations. Notably, there is a striking similarity between the dynamic connectivity in the alanine-dipeptide CSN (Figure 5.7, top) and the optimal free-energy pathways reported in a previous work (see Figure 3 of Ref. [6]). It is worth noting that the network contains the dynamic connectivity whereas the projection of the free energy onto (ϕ, ψ) does not illustrate pathways (Figure 5.7B).

To obtain a quantitative description of the thermodynamics and kinetics of the system, the relation between the cluster structure of the network and the energy basins has been investigated in more details. MCL with a value of 1.2 for the granularity parameter r [172, 61], finds four clusters (represented in red, blue, green and magenta in Figure 5.7A). Each of the C_{7eq}, α_R, C_{7ax} and α_L minimum is grouped into a separate cluster.

In Figure 5.7C cells of the (ϕ, ψ) space are colored according to the clusters found by MCL. Interestingly, the cluster structure reflects very well the architecture of the energy landscape and cluster borders match the saddle points and isoline of the corresponding free-energy projection (see below for the definition of yellow nodes). This result indicates that network clusters retrive the correct free-energy basins of the peptide.

Apart from identifying the main energy basins, the community structure analysis of CSN can reveal as well the transition states. Transition states are the regions connecting different energy basins and correspond to saddle points of the energy landscape. The identification of transition states has been discussed in several pioneering work (see for instance [109, 99, 136, 175]) in structural biology and still remains a crucial challenge. In the framework of CSNs, and assuming that energy basins have been correctly identified by the clustering algorithm, transition states are likely to correspond to *unstable* nodes [61], i.e. nodes lying between two clusters. The stochastic algorithm presented previously in this Thesis (see Chapter 4) has been used in combination with MCL for the detection of unstable nodes in the di-alanine network. Since the network is characterized by a large heterogeneity in the node degree, we used the non-homogeneous way of introducing noise into the network. In this version, the noise over the edge weight is given by $\sigma_{ij} = \pm w_{ij} \left(1 - \frac{1}{\sqrt{\min(k_i, k_j)}}\right)$, with the sign of σ_{ij} chosen randomly for each edge. 100 runs of MCL have been performed on the di-alanine network for different noisy realizations and the threshold θ was set to 0.96 (see Chapter 4). Lower values of θ sample slightly less transition states, while larger values yield spurious results. In Figure 5.7 unstable nodes are colored in yellow. Especially for the well sampled transition $C_{7eq} \leftrightarrow \alpha_R$ unstable nodes characterize the saddle regions of the (ϕ, ψ) space showing that instabilities detection is able to determine inter-basin transition regions without the use of reaction coordinates such as the number of native contacts.

Despite the large amount of noise added to the network (for high degree nodes, $\sigma_{ij} \approx w_{ij}$),

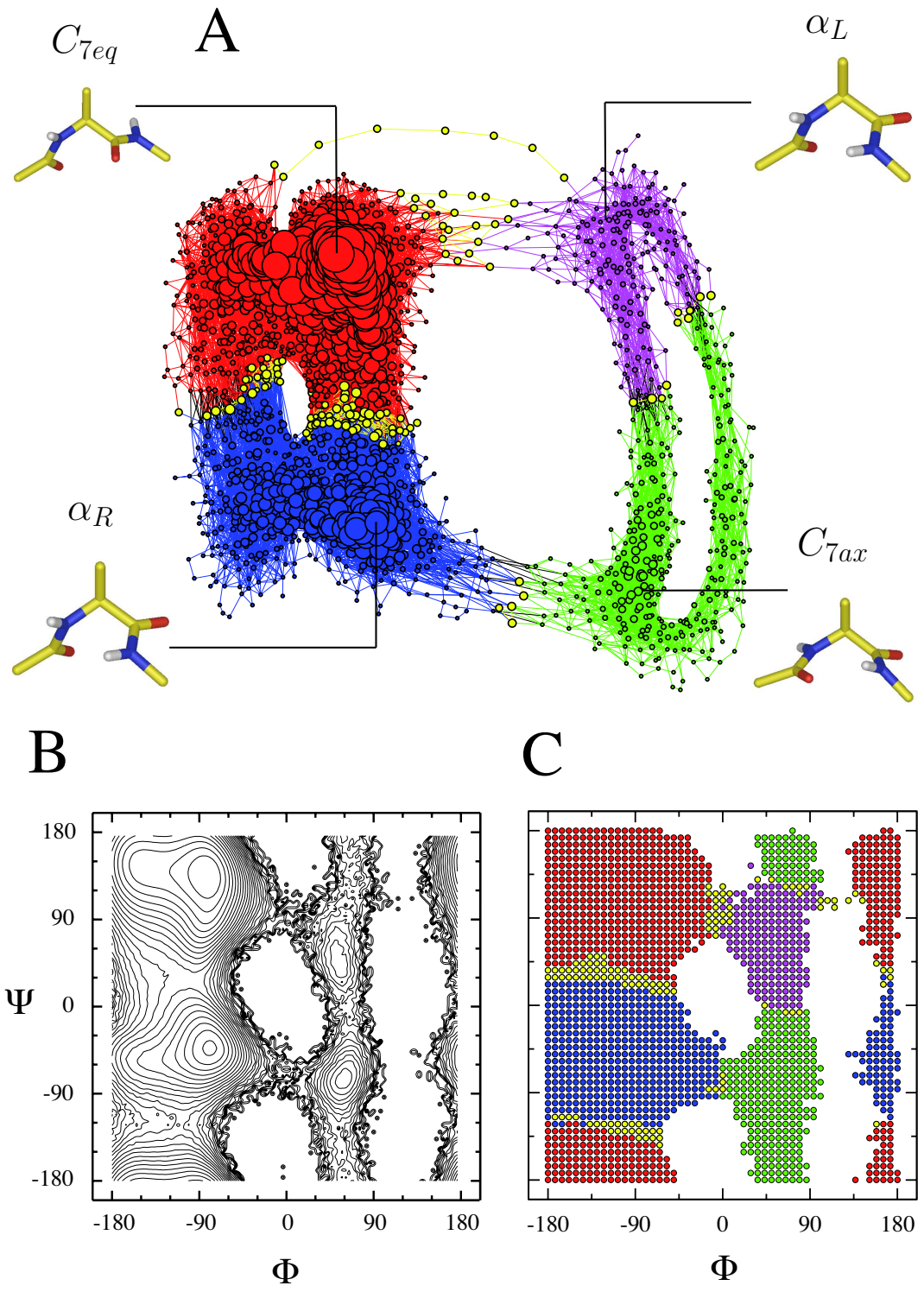


Figure 5.7: Di-alanine peptide free-energy landscape. **A**: Plot of the CSN. Node size is proportional to node weight. **B**: Free-energy projection on the dihedral angles ϕ and ψ. Isolines are drawn every $0.5k_BT$. **C**: (ϕ, ψ) representation of the configurations (nodes) used for building the network. Colors in A and C are set according to the cluster structure found by MCL using $r = 1.2$. Yellow nodes represent unstable configurations identified by the method of Chapter 4. For clarity their size has been slightly increased.

the community structure exhibits a very robust behavior and unstable nodes are very localized. The clustering entropy takes a value of 0.15 (see for instance Table 4.1 for a comparison with the synonymy network). Overall, these results indicate that the community structure of the di-alanine network is very stable and that clusters correspond to a real structure of the network.

5.3.4 Other phase-space discretizations

An important issue while studying CSN is to assess the influence of the phase space discretization into small cells. In Figure 5.7, the phase space of the di-alanine was partitioned into 50×50 cells according to the ϕ and ψ angles. A coarser discretization of the (ϕ, ψ) space in 20×20 cells results in a network of 348 nodes. MCL with $r = 1.2$ detects four communities which are shown in Figure 5.8. The community structure found in this case is very similar to the one of Figure 5.7.

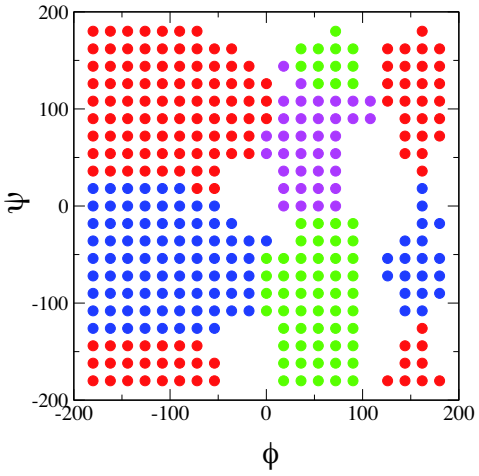


Figure 5.8: (ϕ, ψ) representation of the dihedral discretization in 20×20 cells. Nodes are color coded according to the communities detected by MCL with $r = 1.2$.

A more stringent test on the robustness of free-energy basins detection is carried out considering a completely different discretization of the configuration space based on inter-atomic distances. Each cell of the configuration space is defined by an array of inter-atomic distances of the atoms of the central alanine residue , e.g., (d1, d2, d3, d4, d5, d6, d7, d8, d9, d10) (see Table 5.1). During the simulation the 10 distances have fluctuations between 0.5 Å to more than 2 Å. Inter-atomic distance values are discretized using bins of size $a = 0.3$ Å resulting in a network of 10713 nodes. Here again MCL identifies 4 communities. Projecting the configurations on the (ϕ, ψ) space shows that the community structure resembles strongly the one obtained with (ϕ, ψ) discretization (see Figure 5.9). Therefore clusters found when cells are defined according to atomic distances are coherent with those obtained with dihedral angles.

Other Clustering of the Di-alanine CSN

MCL results show that the community structure of CSN gives a quantitative description of the free-energy basins of a complex system, independently of the phase space discretization scheme. However, correct partition of the network into free-energy basins might depend on the parameter r. In MCL, varying the value of the parameter r changes the granularity of the clusters. To

Table 5.1: List of the 10 inter-atomic distances for the atoms belonging to the central alanine residue

distance name	interacting atoms
d1	N : C_β
d2	N : C
d3	N : O
d4	H : C_α
d5	H : C_β
d6	H : C
d7	H : O
d8	C_α : O
d9	C_β : C
d10	C_β : O

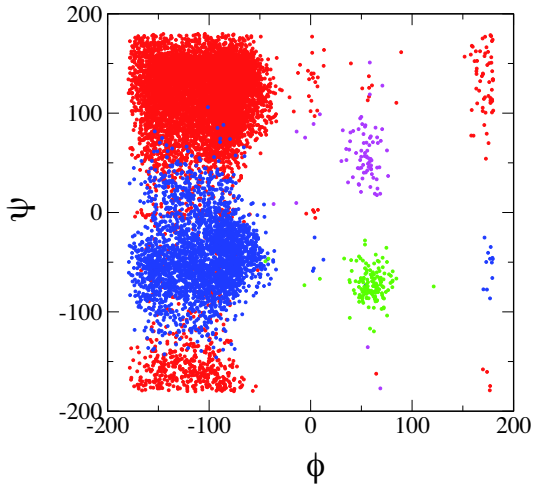


Figure 5.9: (ϕ, ψ) projection of the configurations defined according to the inter-atomic distance discretization with $a = 0.3$ Å (see Table 5.1). Nodes are color coded according to the communities detected by MCL with $r = 1.2$.

check this effect, r has been varied from 1 to 1.5. The resulting community structure of the CSN built from the 50×50 (ϕ, ψ) discretization is shown in Figure 5.10. At $r = 1$ the network consists in one single cluster by construction of MCL. Increasing the value of r results in the splitting of the C_{7eq} and α_R basins first and then C_{7ax} and α_L. Further increase of the value of r leads the detection of the marginally stable C_5 basin (C_5 minimum has $(\phi, \psi) = (-140, 140)$). Interestingly the hierarchy of communities obtained when increasing r is coherent with the ratio f_w between the number of transitions from one community to another and the total weight of this community. f_w is shown in Figure 5.10 for each new community discovered by MCL. In itself this ratio is not informative since it depends on the saving frequency. However comparing f_w for different pairs of communities gives an indication of how strong the communities are as seen from the MD simulations point of view. Equivalently it can be interpreted as a comparison

of the energy barriers between basins.

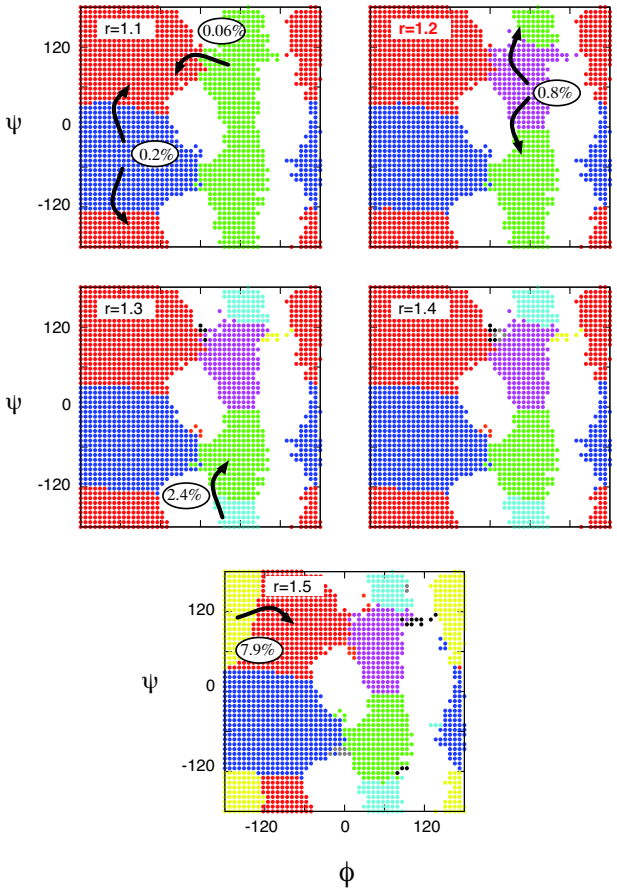


Figure 5.10: (ϕ, ψ) projection of the communities found with MCL and r = 1.1, 1.2, 1.3, 1.4, 1.5. The value within the white ellipses shows the ratio f_w between the number of transitions from one community the another and the total weight of the starting community.

Two other clustering algorithms have also been applied to the CSN of the di-alanine shown in Figure 5.7: Potts clustering [149] and modularity optimization [33] (see Chapter 3). The Potts clustering is a parametric algorithm and the larger the parameter γ is, the more clusters are observed. Figure 5.11 shows the (ϕ, ψ) projections of the clusters identified by the two methods.

The Potts clustering performs well for a range of γ values. This finding is important since it double-checks the validity of the clusters found with MCL. The only slight difference lies in the fact that varying γ does not exactly respect the increase of the ratio f_w between communities.

Modularity optimization has the advantage of not including any parameter. Unfortunately it splits the native basin of di-alanine into three clusters, which does not make sense in this case. Although the modularity of this partition (0.58) is significantly larger than the one of the correct partition with MCL (0.15), clusters do not correspond relevant energy basins. This example shows that modularity optimization, albeit very popular, can lead to erroneous results [54].

The three clustering approaches are very different in spirit with respect to each other and were not developed for any specific network target. The global agreement between MCL and Potts-like algorithm, as well as the low clustering entropy found with MCL, indicate that clusters are not only an artifact of the algorithm. However a completely blind application of a

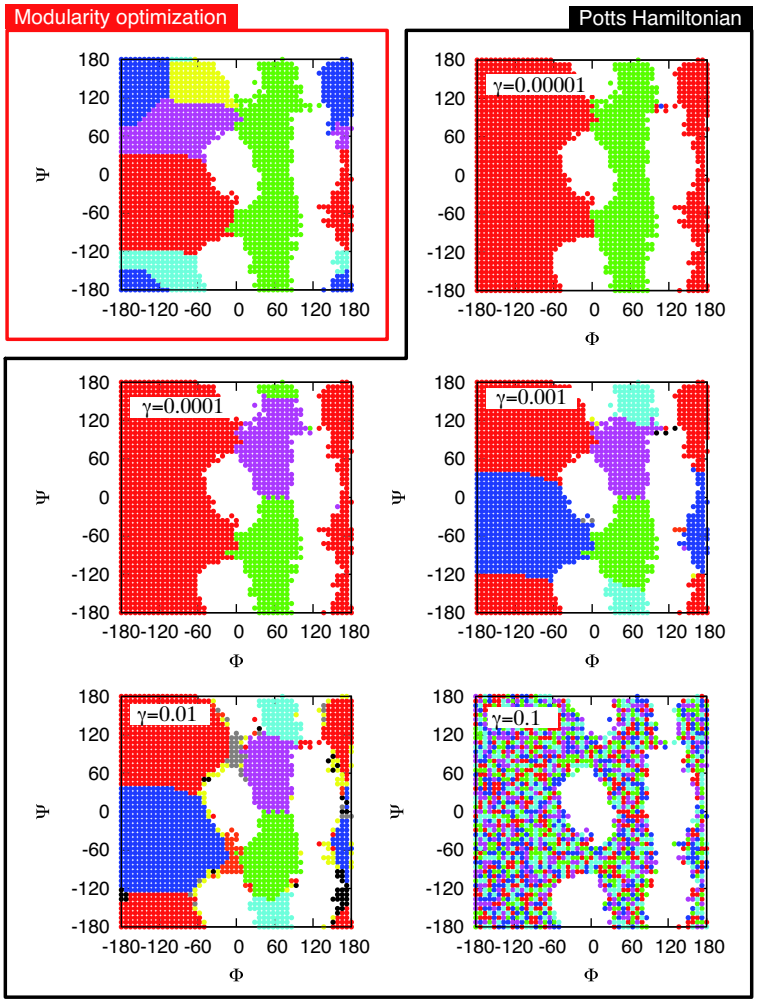


Figure 5.11: (ϕ, ψ) projection of the communities found with both the modularity optimization [33] and the Potts model [149] clustering.

clustering algorithm may give incorrect results, which suggests that some prior knowledge of the characteristics of the system under study is important for a correct interpretation of the network communities.

Finally, given the nature of the problem, MCL appears to be most suitable algorithm for the detection of free-energy basins in CSNs, since it mimics a stochastic exploration, which is the way in which the network was built.

5.4 Conclusion

In conclusion, CSNs provide an interesting example of the use of networks to tackle and reduce the complexity of large and intricate systems. In these networks, the community structure has been shown to correctly reveal the different energy basins, both for simple energy landscapes and for more complex dynamical systems. Identifying the energy basins strongly reduces the complexity of the network and allows to retrieve the higher level of organization by grouping together various configurations. Furthermore the combined use of CSN and community detection

allows for visualization of high-dimensional energy landscape without requiring a projection on one or two arbitrary reaction coordinates. From a more general perspective, the results obtained with simple models energy landscape suggest that the framework of CSN is not restricted to protein folding issues, but might be relevant for any stochastic process.

Finally the analytical derivation and simulations presented in Appendix A provide first rationale for the topology of CSNs, and especially the weight distribution. The main result emerging from this analysis is that heavy tails, and sometimes power-laws, originate from the funnel shape of enthalpic basins, while entropic basins are characterized by a Poissonian weight distribution.

Part II

Coarse graining complex networks

Chapter 6

Introduction

One of the most difficult aspects in the analysis and visualization of complex networks is their large size. In particular size becomes quickly an insuperable hurdle for modeling networks, simulating their dynamical behavior or extracting relevant global informations, since most algorithms used to achieve these tasks run in times that grow polynomially (if not exponentially) with the number of nodes in the network. In this sense even networks of a few thousands nodes can represent a challenge.

A natural way around this problem is to reduce the network complexity by reducing the number of nodes in such a way that the resulting network becomes amenable for analysis and visualization. In order to achieve this goal, most existing techniques (in particular the clustering described in Part I) are based on the idea of either grouping nodes together or removing some nodes. However, to be effective and reliable, complexity reduction techniques should not only yield a smaller network. They should also fulfill the condition that the reduced network bears as much information as possible about the initial one. In other words the reduced network should be representative and keep at least some of the most relevant properties of the original network. This question has often been disregarded in the analysis of complex networks. In particular clustering algorithms described previously in this Thesis only partly reach the goals stated above. Indeed the network size is significantly decreased, but no clear statement is made on whether the network of clusters is representative of the initial one or not. New strategies are therefore needed to reduce the complexity of networks, ensuring at the same time that their relevant properties are preserved.

Complexity reduction techniques satisfying the goal of preserving some (maybe all) properties of a system are often referred to as *coarse graining*, since they provide a reduced system in which some of the fine details have been smoothed over or averaged out. The notion of coarse graining is very common in statistical physics. It has been extensively used to study phase transitions and critical phenomena, and stands at the heart of the renormalization group. Moreover another kind of coarse graining is often performed in the field of machine learning and artificial intelligence using Principle Component Analysis. Despite the success of the coarse graining approach in these fields, coarse graining complex networks has received little attention up to now. The present work aims at filling this gap in the analysis of complex networks.

As a final remark, we note that reducing the network complexity most often loses some information about the initial network. A crucial issue is therefore to decide which properties should be preserved. On the one hand topological properties such as the degree distribution or correlations might be important to preserve, because of their effect on the network architecture. On the other hand not all interesting features of a network are described by the topology and network dynamics is known to play a key role as well [18, 180]. In this Part we use the fact that the relevant properties concerning networks and their dynamics are often encoded in some

particular eigenvalues and eigenvectors of a matrix describing the network. Focusing on these eigenvectors, we show that there exists a natural way of grouping nodes so that the properties of interest are preserved in the reduced network. For this reason we refer to our new method as *Spectral Coarse Graining* (SCG).

Outline

We first review in Section 6.1 the few existing approaches to coarse grain complex networks. Then, in Chapter 7, we introduce the mathematical framework to coarse grain networks based on the spectral properties of the stochastic matrix. Several applications of the Spectral Coarse Graining are presented in Sections 7.4, 7.5 and 7.6 involving different real world networks. Finally a complete discussion of the method is provided in Section 7.7. In a second stage (Section 7.8), the method is extended to account for symmetric matrices such as the adjacency or the Laplacian matrix. As an application we consider the Gaussian Network Model describing fluctuations from equilibrium in proteins or polymers. Eventually conclusions and perspectives are drawn in Section 7.9.

6.1 State of the art

The idea of coarse graining is a key concept in statistical physics. Its importance stems from the fact that systems and models described by the laws of statistical mechanics are composed of a tremendous number of atoms, typically scaling as $N_A \sim 10^{23}$. While any calculation becomes completely unfeasible even with the most powerful computers available nowadays, it has been observed already long ago that several of those systems are equivalent to smaller ones, i.e. have the same partition function. For instance the partition function of a spin system on a 1D-lattice is equivalent to the one in which every third node has been removed, provided that interactions are suitably redefined [34]. For such systems the coarse graining can be iterated infinitely. This invariance has dramatic consequences and characterizes systems exhibiting critical behavior, which have been described in the framework of the renormalization group.

Because of the highly heterogeneous structure of complex networks, nothing comparable to renormalization group in statistical physics could be defined. For this reason, most physicists working in the field of complex networks have been focusing on the problem of community detection (see Part I) as a way to reduce the network complexity. Yet, a few attempts have been made to define a coarse graining scheme for complex networks, most often considering the topology of the network. In parallel scientists working in computer sciences and machine learning have also proposed some interesting approaches related to the goals of coarse graining networks.

Here we review the existing approaches related to the coarse graining of complex networks. Most of them are based on local properties of networks, in particular the degree of the nodes. With respect to these approaches, the SCG scheme introduced in Chapter 7 represents an important shift in the way to address the problem of reducing the network complexity.

6.1.1 k-core decomposition

The k-core decomposition, which is a node decimation technique, was first proposed in [162] and [20] to isolate the central core of a network. It is based on the idea that highly interconnected nodes play a central role in the network. A k-core is defined as a maximal subgraph, not necessarily connected, such that all nodes of the subgraph have at least k edges pointing to other nodes within the subgraph. k-cores can be identified by a simple procedure. First all nodes with degree lower than k are removed. After the removal, some nodes might have a new degree lower or equal to k. They are further removed and the procedure is iterated until all nodes

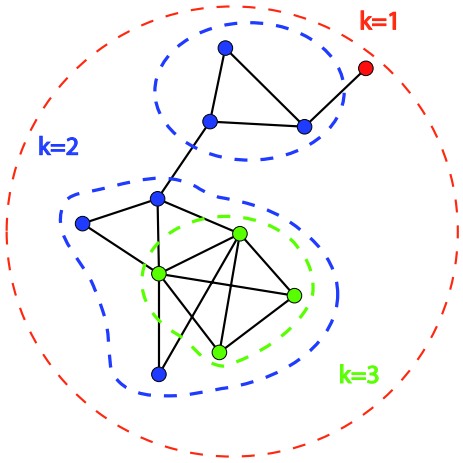


Figure 6.1: Sketch of the k-core decomposition for a small network.

have a new degree larger than k. These nodes form the k-core of a network. Depending on the network and on k, a k-core may be a disconnected subgraph (Figure 6.1). Finally if the network is strongly disassortative (i.e. nodes of high degree are connected only to nodes of low degree), the k-core becomes empty already for small values of k. For this reason, k-core decomposition is likely to provide more insights for assortative or uncorrelated networks than for disassortative ones.

Indeed every k-core is included in the $(k-1)$-core and the k-core decomposition results in a network organization similar to a Russian nesting doll or an onion shell. Recently a complete characterization of k-core organization in uncorrelated networks was derived in [39].

From a coarse graining point of view, the k-core decomposition of a network preserves the highly intra-connected nodes. As the name indicates, it allows to identify the central parts of a network (the central cores), no longer considering peripheral nodes (small degree) or large degree nodes connected only to low degree ones. For these reasons k-core decomposition has been used as a visualization tools in which one can zoom into a network following the different levels of organization induced by the k-cores [3].

Finally k-core decomposition differs from a clustering approach in the sense that it aims a providing a reduced network in which all nodes sufficiently connected to each other are preserved and others are removed.

6.1.2 Box-counting

The use of box-counting techniques to coarse grain complex networks was first introduced by Song *et al.* [165] and further analyzed by Goh *et al.* [68], but the underlying ideas go back to the work about fractals and self-similarity under a renormalization procedure [102, 176]. Fractal systems are characterized by a structure which looks the same at all length scales. To unveil the different length scales of a system, a common approach is based on grouping the system units into boxes, the size of the boxes determining the length scale at which the system is observed. Box-counting techniques are most easily understood for a system living in Euclidean space, i.e. for which a distance can be defined between the different units. The space is covered by boxes of linear size l and the units falling in the same box are considered as a new single unit. In fractals systems the number of boxes N_B required to cover the space and their linear size l are related by the following equation:

$$N_B \propto l^{-d}, \tag{6.1}$$

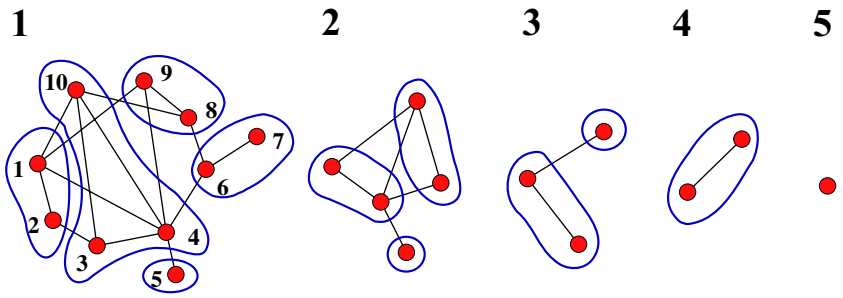


Figure 6.2: Example of five different stages in the coarse graining scheme of Song *et al.* with $l = 2$ for a small network of 10 nodes.

with d the fractal dimension of the system. In order to apply the box-counting approach to complex networks the notion of geographical distance was extended to the distance between the nodes of a network [165], defined as the shortest path length. A box of size l typical contains nodes separated by a distance smaller than l. Figure 6.2 shows how box counting with $l = 2$ performs on a small network. First node 1 and 2 are grouped. No other node can belong to this box since they would have a distance of at least 2 with one of the two nodes. Other boxes are built similarly and finally the system is tiled by considering each box as a node at the larger scale defined by l.

As it can be seen from Figure 6.2 there exist several ways to cover a network with boxes of size l. By exhaustive searching, Song *et al.* found the covering requiring the fewest boxes, and then examined how the number of boxes N_l depends on l. They found that many (but not all) real-world networks follow the power-law scaling of Eq. (6.1) just as if they were fractal shapes. In addition they have shown that the degree distribution remained often constant along a few coarse graining stages and for different choices of l. From these observations a theory of self-similarity in complex networks has been elaborated [165, 166].

At this point it should be stressed that not all complex networks exhibit a "fractal" organization. And naturally the question arises whether the observed fractality stems from a self-organization principle leading to a critical behavior, such as systems characterized by a phase transition, or is simply a artifact of the box-counting applied to some networks [168]. Up to now no satisfactory answer has been found to this question.

Albeit the lack of clear interpretation, the results of Song *et al.* have pointed out that network reduction should go hand-in-hand with the preservation of some relevant network properties, akin to the coarse graining in statistical physics. Their work focuses on the degree distribution. In the SCG approach, the features of interest will be the spectral properties of networks.

6.1.3 Geographical coarse graining

In 2004, Kim introduced a coarse graining scheme for geographical networks embedded on a square lattice [86]. This work was motivated by the recent studies about brain networks. The ultimate goal of a global approach to human brain organization is to have a complete cartography of the neurons and their physical connections or correlations. Unfortunately this is nowadays completely unfeasible because of the tremendous number of neuron connections ($\approx 10^{15}$). To simplify the problem, a common approach is to partition the brain into cubic cells (voxels) and to measure the activity correlation between the different cells [46]. Typically hundred to thousand voxels are defined over the entire brain. In this way, a weighted network can be constructed in which nodes are voxels and edge weight represents correlation in the activity observed between two voxels. The network topology is based on heavily coarse-grained

information and a key issue is to know how relevant the properties of such networks are to the original system. In [86] a model has been designed to study the effect of coarse graining. In this model the nodes of the initial network are located on a 2D-lattice. Then a degree is assigned to each node from a given distribution (typically a power-law distribution) and the connections are drawn minimizing the length of the edges (see [86] for more details). The coarse graining scheme resembles the Kadanoff block-spin renormalization group approach: at each stage the four nodes forming 2×2 squares on the lattice are merged together summing up all their edges. It is further required that the average degree remains the same (which is obtained by removing parts of the low-weight edges). Applying the coarse graining procedure to networks with a power-law degree distribution, Kim has shown that the scale-free topology was preserved and the exponent remained constant under several stages of the coarse graining.

Although the coarse graining scheme was applied for a 2D-lattice (while the brain is a 3D system) and requires to remove some edges along the coarse graining to ensure that the network does not become a complete graph, this work indicates that networks whose nodes are embedded in an euclidean space may be coarse-grained without changing significantly their topology.

6.1.4 Ring Structure

In [13], a slightly different point of view was adopted. Instead of focusing on the topology of the network, in particular the degree distribution, Baronchelli *et al.* have shown that the mean first passage time to a node in a network can be computed by considering a much reduced graph reflecting the ring structure of the network. Given a node i, the ring l includes all nodes at distance l from i. The reduced network is built by identifying each ring as a node, summing up all the edges. Its structure is the one of a chain with a length smaller or equal to the diameter of the network. Remarkably, it has been shown in [13] that the mean first passage time computed in the reduced graph is exact for random graphs and provides an excellent approximation for other kinds of network.

The first goal of Baronchelli *et al.* was to design a framework for computing the mean first passage time in a faster way. More generally their work points to the important observation that network topology, in particular the degree distribution, might not be the only relevant feature that should be preserved under coarse graining.

6.1.5 Clustering

Clustering algorithms are powerful complexity reduction techniques allowing to significantly decrease the size of a network by clumping nodes into groups or communities (see Part I of this Thesis). After grouping the nodes a much reduced "network of clusters" is obtained. However most clustering approaches disregarded the question of which properties of the initial network are preserved in the network of clusters (an exception can be found in [137] in which the degree distribution in the network of clusters has been studied). Therefore clustering cannot be considered as a proper coarse graining approach since the ultimate goal of a clustering algorithm is to find the "correct" communities of a network and not to ensure that the network of clusters leaves unchanged some properties of the initial one.

Further in this thesis, we will show that there exists a connection between clustering and Spectral Coarse Graining, though. However for most real networks characterized by a fuzzy community structure, both preserving some properties of the network and finding reasonable communities are often irreconcilable tasks. This observation combined with the enthusiasm generated by community detection methods in the field of complex networks can partly explain while coarse graining complex networks has received so little attention among scientists working in this field.

More relevant are the works of computer scientists and mathematicians, with respect to the connection between clustering and coarse graining. In [110] Meila and Shi have designed a spectral clustering technique, referred to as the Normalized Cut, based on the stochastic matrix W. Although their goal is not to coarse grain networks, in the sense we have defined it, an interesting result can be found in the Appendix of their work. First they define a *piecewise constant* vector as a vector whose components are constant over each group of a partition of the network. Considering a partition such that the n first left eigenvectors of the stochastic matrix are piecewise constant, they show that this particular partition (as well as any other sub-partition) results in preserving the n first eigenvalues in the reduced network. This result will be reinterpreted and extended in Chapter 7.

In another work, Lafon and Lee [95] have recently studied the consequences of grouping nodes into clusters on the behavior of random walks. They consider the general case of any partition of the network into $\tilde{N}$ groups. Given the partition, nodes can be merged within each group, summing all their edges, to form a reduced network. On the reduced network, a random walk is described by a $\tilde{N} \times \tilde{N}$ stochastic matrix $\tilde{W}$. For any right eigenvector of the stochastic matrix W, a reduced eigenvector of size $\tilde{N}$ can be defined in which the components of each group are summed. In [95], Lafon *et al.* have shown that the difference between the reduced eigenvectors of W and the corresponding eigenvectors of $\tilde{W}$ has an upper bound depending on the partition. It has been suggested that finding a partition minimizing the upper bound could provide an effective clustering algorithm. In addition it would allow to preserve as much as possible the eigenvalues and eigenvectors of the network, which is exactly the aim of a consistent coarse graining technique. A related method can also be found in [60].

In our approach, a similar mathematical framework will be designed. In addition we will show that there exists a very intuitive way to define a partition on the network such that at least some eigenvalues and eigenvectors of the stochastic matrix are preserved in the reduced network.

6.1.6 Principle Component Analysis

Principle Component Analysis (PCA) is a well-known technique to reduce the dimensionality of a data set where each entry is given as a vector $|x_i\rangle$ $(i = 1, \ldots, N)$ in a D-dimensional space [84]. PCA is based on the idea of expressing the data in a new orthogonal basis such that the covariance is maximal along the first basis vectors (see Figure 6.3) and the covariances along the directions defined by the eigenvectors of C are equal to the eigenvalues of C. In practice, principle components are obtained as the eigenvectors $|p^\alpha\rangle$ of the covariance $D \times D$ matrix $C_{lk} = \frac{1}{N-1} \sum_{i=1}^{N} x_{i_l} x_{i_k}$. Eigenvectors corresponding to the largest eigenvalues are referred to as *principle* since they have been shown to span the subspace corresponding to the maximal covariance of the data set.

PCA is especially useful if some coordinates (say l and k) turn out to be somehow (linearly) correlated. In this case a few eigenvalues of the matrix C will be much larger than the others. This finding implies that the most relevant information in the data set is contained in the first eigenvectors of C. For this reason PCA can be used as a dimensionality reduction technique since projecting the data along the principle components allows to reduce the dimensionality of the data set without losing the most relevant information. Mathematically this projection is equivalent to truncating the spectral decomposition of the vector $|x_i\rangle = \sum_{\alpha=1}^{D} \lambda^\alpha a_i^\alpha |p^\alpha\rangle \approx \sum_{\alpha=1}^{\tilde{D}} \lambda^\alpha a_i^\alpha |p^\alpha\rangle$, where $\tilde{D}$ is the number of principle components. It is important to stress that in PCA the number of data points (N) remains the same, only the dimension of the data set is reduced $(D \to \tilde{D})$. For this reason PCA differs from clustering technique, although both approaches have sometimes been combined [187].

A few attempts have been designed to extend PCA to networks [156]. Considering the adjacency matrix A (or any other matrix such as the Laplacian L or the stochastic matrix

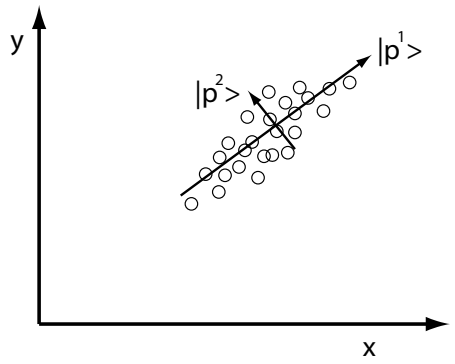


Figure 6.3: Example of PCA. Data points are given by N vectors in a 2-dimensional space. Principle component $|p^1\rangle$ gives the direction of maximal covariance.

W), one could naively try to apply the same projection writing $A = \sum_{\alpha=1}^{N} \lambda^\alpha |p^\alpha\rangle\langle p^\alpha| \approx \sum_{\alpha=1}^{\tilde{N}} \lambda^\alpha |p^\alpha\rangle\langle p^\alpha| = A'$. However this approach leads to spurious results. In particular negative entries appear in A', indicating that A' is no longer an adjacency matrix. In Chapter 7, we will show that there exists a way to project the matrix A on its principle components such that the resulting matrix still accounts for a (reduced) network. Furthermore the size of the network is significantly decreased, whereas the "direct" PCA projection yields a $N \times N$ matrix A'. For this reason Spectral Coarse Graining [63] (see Chapter 7) can be reinterpreted as an extension of Principle Component Analysis for networks combined with a size reduction.

6.1.7 Miscellaneous

Many other coarse graining schemes have been designed to simplify specific networks. For instance simulations of large protein dynamics have extensively used the idea of coarse graining. Instead of taking into account all atoms, it has been shown that accurate results could already be obtained considering only the C_α (structural coarse graining). Moreover atoms have been represented in first approximation as beads interacting through a network of springs. Considering each atom as a node of a network, the removal of all non-C_α atoms is equivalent to coarse graining the network, taking advantage of the prior knowledge about the role and importance of each node. Various similar works have focused on coarse graining the structure of proteins to study them with the help of simplified models [178, 67, 77, 105, 192, 31] ([31] shares some similarity with Spectral Coarse Graining, although the goals and the results are quite different). In a more advanced coarse graining method, the phase behavior of proteins was even shown to be well-described by a flexible tube model [104, 10].

In a completely different field, PageRank quantity [24] has been approximated with high precision by considering a coarse graining of the Web [85] based on the address of each web page. A local PageRank is first computed for pages of the same host, and then a global PageRank is defined considering each host as a node of the coarse-grained network.

Since most of these approaches are case-dependent and often rely on specific *a priori* informations about the nature of the nodes, describing all of them will likely not give new insights to the reader regarding a global approach to network coarse graining. We therefore restricted ourselves to a few cases of particular interest, some of them being related to examples presented in the next Chapter. As a concluding remark, we believe that the vast amount of complex systems that need to be coarse grained to study them, and at the same time are well-described

in the framework of complex networks, is a strong indication that if a global coarse graining scheme can be defined, coarse graining complex networks is a relevant starting point. We hope that our work might contribute to this task and we now turn to the description of the Spectral Coarse Graining recently introduced [63].

Chapter 7

Spectral Coarse Graining

In this Chapter the mathematical framework underlying Spectral Coarse Graining (SCG) is introduced in terms of stochastic matrices describing random walks on networks. The restriction to stochastic matrices has two main reasons. First it allows for a direct and very natural relation between the idea of grouping nodes and the one of preserving the spectral properties of a network. Second the coarse graining was first implemented in this framework and most results presented in this Thesis have been obtained with stochastic matrices [63]. Section 7.8 deals with an extension to symmetric matrices (either the adjacency or the Laplacian matrix), for which most results obtained with stochastic matrices are valid.

The Chapter is organized as follows. In Section 7.1 we review some properties of random walks and stochastic matrices. In Section 7.2 the framework for an exact SCG is described in details. We further prove analytically in Section 7.3 that a perturbation approach can be carried out. This is the most important result on which efficient SCG of complex networks relies. Spectral Coarse graining is then applied to di-alanine network (Sections 7.4 and 7.5) and to a periodic gene network in Section 7.6. Finally we discuss the parameters involved in SCG and the connection with spectral clustering algorithms in Sections 7.7.

7.1 Motivations

Random walks and diffusive processes play a key role in the dynamics of a large number of complex networks [133]. For instance random exploration of networks such as the web crawls of search engines are described by random walks. Diffusive processes on networks appear also in traffic simulation, percolation theory or message passing processes. More generally Markov Chains can be considered as random walks on a network in which nodes correspond to the different states of the system. Finally in Chapter 5 we have shown that even continuous diffusive processes can be mapped onto a network and random walks on this network represent the system dynamics.

The evolution of a random walk is described by the stochastic matrix W defined as $W_{ij} = A_{ij} \left(\sum_l A_{lj}\right)^{-1}$. For connected and undirected networks, the eigenvalues of W satisfy: $1 = \lambda^1 \geq \lambda^2 \geq \ldots \geq \lambda^N$

Because of the column-normalization, W is not symmetric and eigenvectors have to be distinguished between left and right eigenvectors. Right eigenvectors are directly related with the evolution of the probability of being in a given node. The right eigenvector $|p^1\rangle$ corresponding to λ^1 is the stationary state. Moreover right eigenvectors with eigenvalues close to one capture the large-scale behavior of the random walk, whereas eigenvectors with smaller eigenvalues contain the small-scale behavior. This can be seen considering the spectral decomposition of the probability density vector $P(n)$ of a random walk over the nodes of a network at discrete time

n. $P(n)$ evolves from the initial distribution $P(0)$ to the stationary state $P(\infty)$, assuming that this state exists, through a transient that can be decomposed over the set of right eigenvectors $|p^\alpha\rangle$ of the stochastic matrix W as

$$P(n) = \sum_{\alpha=1}^{N} a^\alpha \left(\lambda^\alpha\right)^n |p^\alpha\rangle \tag{7.1}$$

where a^α is the projection of the initial distribution over the α^{th} eigenstate. Up to a normalizing constant, $|p^1\rangle$ is equal to the stationery state $P(\infty)$ of the random walk. For simple networks, $|p^1\rangle$ is given by $p_j^1 = \sum_i A_{ij} = k_j$. Eq. (7.1) shows that the largest eigenvalues dictate the behavior of the random walk at large time scales.

Left eigenvectors on the other hand have been shown in Chapter 2 to appear in spectral clustering algorithms (they are equal to the right eigenvectors of the normal matrix). Starting from the definition of $|p^\alpha\rangle$, we have for undirected networks:

$$\begin{aligned} \lambda^\alpha p_j^\alpha &= \sum_i \frac{A_{ji}}{p_i^1} p_i^\alpha \\ \Leftrightarrow \lambda^\alpha \frac{p_j^\alpha}{p_j^1} &= \sum_i \frac{A_{ji}}{p_j^1}\frac{p_i^\alpha}{p_i^1} = \sum_i \frac{A_{ij}}{p_j^1}\frac{p_i^\alpha}{p_i^1} \end{aligned}$$

which is exactly the equation for the left eigenvector $\langle u^\alpha|$. Thus,

$$u_i^\alpha = \frac{p_i^\alpha}{p_i^1} \propto \frac{p_i^\alpha}{\sum_j A_{ji}} \tag{7.2}$$

Eq. (7.2) shows that left eigenvectors are equal to right eigenvectors rescaled by the degree of the nodes. Because of the column normalization $\langle u^1| = (1, 1, \ldots, 1)$, which indicates that right eigenvectors bear the fingerprint of the node degree, while left eigenvectors do not. For this reason (and many others that will become clear as we describe the Spectral Coarse Graining), it is natural to consider left eigenvectors in a coarse graining procedure.

Finally we stress that Eq. (7.2) is only valid for undirected networks. For simplicity we assume undirected networks in the next Section. Whenever the generalization to directed networks is not valid, we will simply indicate it and refer to Section 7.3.3 for a discussion about directed networks.

7.2 Exact coarse graining

As a starting point for a coarse graining strategy, we want to ensure that two nodes (say nodes 1 and 2) having exactly the same neighbors are grouped together, since they cannot be distinguished from the point of view of a random walk starting anywhere else in the network. This is indeed an ideal case which might not lead to a significant size reduction of the network. In a second stage (Section 7.3), we will show analytically that the ideal case can be extended by carrying out a perturbation approach.

Figure 7.1 illustrates the ideal case. The two green nodes have exactly the same neighbors, which implies that columns 1 and 2 of the stochastic matrix are equal, $W_{i1} = W_{i2}$. In terms of a left eigenvector $\langle u^\alpha|$ of W it means that $u_1^\alpha = u_2^\alpha$ for $\lambda^\alpha \neq 0$ [1].

The obvious coarse graining step is to coalesce nodes satisfying $u_1^\alpha = u_2^\alpha$, with the resulting new node carrying the sum of the edges of the original ones. The new network in which nodes 1

[1] Note that the converse is not always true: nodes with $u_1^\alpha = u_2^\alpha$ for a given α do not always have exactly the same neighbors. This pathological situation is treated in details in Appendix B.1, but does not change the results presented below.

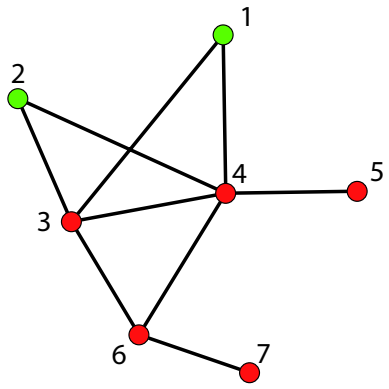


Figure 7.1: Small network presenting the ideal case for SCG. The green nodes have exactly the same neighbors and can be merged without changing the spectral properties of the network.

and 2 have been merged is characterized by a $(N-1)\times(N-1)$ adjacency matrix $\tilde{A}$, with the first line, resp. column, of $\tilde{A}$ being the sum of the two first lines, resp. columns, of A. On this reduced network, the stochastic matrix $\tilde{W}$ describing a random walk is obtained by normalizing the columns of $\tilde{A}$. At this point, it will be useful to write $\tilde{W}$ as a product of three matrices,

$$\tilde{W} = RWK.$$

K and R are two projection-like operators from the N-dimensional space of the initial nodes to the $(N-1)$-dimensional space of the new nodes. In order to fulfill the definition of $\tilde{W}$ and using that $p_j^1 \propto \sum_i A_{ij}$ for undirected networks, K and R are defined as:

$$K = \begin{pmatrix} \frac{p_1^1}{p_1^1+p_2^1} & 0 & \dots & 0 \\ \frac{p_2^1}{p_1^1+p_2^1} & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & I_{N-2} & \\ 0 & & & \end{pmatrix} \quad \text{and} \quad R = \begin{pmatrix} 1 & 1 & 0\dots 0 \\ 0 & 0 & \\ \vdots & \vdots & I_{N-2} \\ 0 & 0 & \end{pmatrix} \tag{7.3}$$

The interesting features of $\tilde{W}$ come from the property [110] that *if* $u_1^\alpha = u_2^\alpha$, *the vector* $\langle u^\alpha|K$ *is a left eigenvector of* $\tilde{W}$ *with eigenvalue* λ^α (i.e. $\langle u^\alpha|K = \langle \tilde{u}^\alpha|$). To obtain this result one simply needs to see that $\langle u^\alpha|KR = \langle u^\alpha|$ if $u_1^\alpha = u_2^\alpha$. Then $\langle u^\alpha|K\tilde{W} = \langle u^\alpha|KRWK = \lambda^\alpha\langle u^\alpha|K$.

For undirected networks we have seen that left and right eigenvectors of W are related by

$$u_i^\alpha = \frac{p_i^\alpha}{p_i^1}$$

Assuming that $u_1^\alpha = u_2^\alpha$, it follows that $KR|p^\alpha\rangle = |p^\alpha\rangle$. Using this property, our result can be extended to the right eigenvectors. Under the same hypothesis ($u_1^\alpha = u_2^\alpha$), *the vector* $R|p^\alpha\rangle$ *is a right eigenvector of* $\tilde{W}$ *with eigenvalue* λ^α. In general this is not true for directed networks. Nevertheless it always holds for $\alpha = 1$, ensuring that the stationary probabilities sum up under the coarse graining.

To summarize we have shown that grouping nodes with the same components in $\langle u^\alpha|$ has a spectral interpretation: it preserves the eigenvalue λ^α, averages the components of $\langle u^\alpha|$ and for undirected networks sums up the components of $|p^\alpha\rangle$.

For simplicity the case where only two components of an eigenvector are equal (resp. close to each other) has been considered. It is straightforward to generalize the grouping to all nodes having the same components (resp. components close to each other) in $\langle u^\alpha|$. Groups are

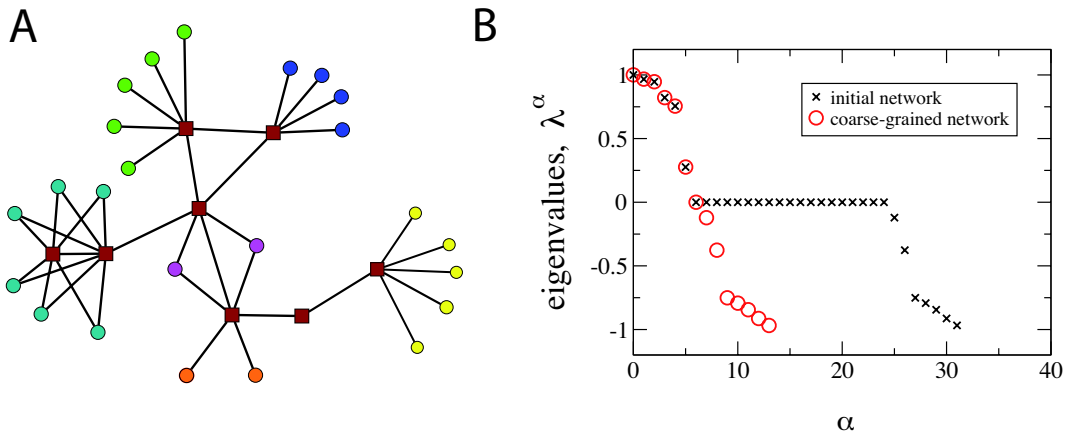


Figure 7.2: Effect of the exact coarse graining on the spectrum of a small network. **A**: Toy network with many nodes satisfying the exact coarse graining. **B**: Ordered list of eigenvalues λ and $\tilde{\lambda}$.

first labeled from 1 to $\tilde{N}$ and $\delta_{C,i}$ is defined as 1 if node i belongs to group C, 0 otherwise ($C = 1 \dots \tilde{N}$). Then K and R read:

$$R_{Ci} = \delta_{C,i} \quad \text{and} \quad K_{iC} = \frac{p_i^1}{\sum_{l \in C} p_l^1} \delta_{C,i}$$

with R a $\tilde{N} \times N$ matrix and K a $N \times \tilde{N}$ matrix, $\tilde{N}$ being the number of different groups.

Before going further in the description of the coarse graining, a few comments are necessary. First of all we stress that the only constraint about the choice of α to satisfy the ideal case is that λ^α should be different than zero. This condition ensures that $u_1^\alpha = u_2^\alpha$ if node 1 and 2 have exactly the same neighbors. Therefore in the ideal case exemplified in Figure 7.1, $u_1^\alpha = u_2^\alpha \ \forall \alpha$ such that $\lambda^\alpha \neq 0$. The ideal case is also characterized by an eigenvalue $\lambda^\alpha = 0$ since columns 1 and 2 of W are the same. These two observations combined with the property that eigenvalues are preserved under coarse graining show that all eigenvalues different than 0 are automatically preserved. As a consequence for any merging of two nodes with exactly the same neighbors an eigenvalue equal to 0 is removed from the spectrum in the coarse-grained network, since the total number of eigenvalues is equal to the network size. Figure 7.2 shows a comparison between the spectrum of W and $\tilde{W}$ of a network characterized by several nodes having the same neighbors. Eigenvalues equal to 0 do not bear any information on a matrix since they do not contribute to the spectral decomposition $W = \sum_\alpha \lambda^\alpha |p^\alpha\rangle\langle u^\alpha|$. For this reason the coarse-gaining restricted to nodes satisfying $u_i^\alpha = u_j^\alpha$ can be considered as exact, in the sense that the entire information of the original network is present in the reduced network and only redundant information has been dropped.

The second comment concerns the matrix K. A few calculations show that the only condition that non-zero entries of K should fulfill such that the properties about eigenvalues and left eigenvectors are valid is the column normalization, which is equivalent to $RK = I_{\tilde{N}}$. Thus other ways of defining K might be designed. This property will prove to be very useful when coarse graining directed networks in which some nodes have a zero stationary probability (see Section 7.3.3). However, if both left and right eigenvectors are to be preserved, the only choice for K is the one in Eq. (7.3). This is the reason why Spectral Coarse Graining is most naturally defined as we did in Eq. (7.3).

Finally the strict equality $u_1^\alpha = u_2^\alpha$ is not often encountered in various kinds of real networks.

For these networks, restricting the coarse graining to the ideal case does not allow to reduce significantly the size of the network and makes the coarse graining approach of little use. To extend the range of our procedure, two natural questions arise:

1. Is it possible to relax the condition $u_1^\alpha = u_2^\alpha$ and to group nodes if u_1^α is close to u_2^α, without requiring the strict equality?

2. In this case, does the reduced network still have the same spectral properties?

These questions are crucial and raise several important issues. The condition $u_1^\alpha = u_2^\alpha$ was automatically fulfilled for all eigenvectors with eigenvalues different than zero in the ideal case. However nothing ensures that if u_1^α is close to u_2^α for a particular α, it will be true for all α (actually it is indeed not true). Therefore relaxing the strict equality implies selecting some eigenvectors along which the coarse graining has to be done. Having in mind the idea of preserving the slow modes of random walks, the natural choice is to coarse grain a network according to the S first non-trivial eigenvectors, i.e. to group nodes having similar components over a set of left eigenvectors $\{\langle u^\alpha|\}_{\alpha=2}^{S+1}$. Yet, before all, it remains to answer the two questions stated above.

7.3 Perturbation approach

Here we show analytically that if groups are chosen such that components of $\langle u^\alpha|$ are almost the same within each group, the eigenvalue λ^α as well as the eigenvectors $\langle u^\alpha|$ and $|p^\alpha\rangle$ are almost preserved in the reduced network. For this reason the following perturbation approach will be referred to as *almost-exact coarse graining*. Several results presented in this Section have been derived by David Morton de Lachapelle considering symmetric matrices and we are thankful to him for enlightening discussions.

Mathematically, it is convenient to express the vector $\langle u^\alpha|$ as:

$$\langle u^\alpha| = \langle u^\alpha|KR + \langle \epsilon^\alpha| \tag{7.4}$$

with $\langle \epsilon^\alpha|$ a N-dimensional vector characterizing the deviation of each component from its weighted average within each group ($\langle u^\alpha|KR$). Indeed if groups have been defined such that components of $\langle u^\alpha|$ are close to each other, all ϵ_i^α are small, i.e. "scale as ϵ^α". If we apply successively W and K in Eq. (7.4), we obtain:

$$\lambda^\alpha \langle u^\alpha|K = \langle u^\alpha|K\tilde{W} + \langle \epsilon^\alpha|WK, \tag{7.5}$$

which immediately shows that for small ϵ^α, $\langle u^\alpha|K$ becomes an approximation of a left eigenvector of $\tilde{W}$. The result can be extended to right eigenvectors $|p^\alpha\rangle$ if the network is undirected since $D|u^\alpha\rangle = |p^\alpha\rangle$, with D a diagonal matrix such that $D_{ii} = k_i$. Using that KRD and D are symmetric matrices, we can evaluate

$$\left(|p^\alpha\rangle - KR|p^\alpha\rangle\right)^T = \langle u^\alpha|D - \langle u^\alpha|DR^TK^T = \langle u^\alpha|D - \langle u^\alpha|KRD = \langle \epsilon^\alpha|D,$$

which gives after applying W and R:

$$\Leftrightarrow \lambda^\alpha R|p^\alpha\rangle = \tilde{W}R|p^\alpha\rangle + RWD|\epsilon^\alpha\rangle,$$

Furthermore we show that an upper bound can be established for $||\langle \epsilon^\alpha|WK||$ in Eq. (7.5). First of all we note that $||\langle \epsilon^\alpha|WK||^2 = \sum_{C=1}^{\tilde{N}} \langle \epsilon^\alpha|(WK)_{\bullet C}\rangle^2 \leq \sum_{C=1}^{\tilde{N}} ||\epsilon^\alpha||^2 \, ||(WK)_{\bullet C}||^2$. Let us now find an upper bound for $||(WK)_{\bullet C}||^2$:

$$\begin{aligned}
\sum_{C=1}^{\tilde{N}} ||(WK)_{\bullet C}||^2 &= \sum_{C=1}^{\tilde{N}}\sum_{j=1}^{N}((WK)_{Cj})^2 \\
&= \sum_{C=1}^{\tilde{N}}\sum_{j=1}^{N}\left(\sum_{l=1}^{N} W_{jl}K_{lC}\right)^2 \\
&\leq \sum_{C=1}^{\tilde{N}}\sum_{j=1}^{N}\sum_{l=1}^{N} K_{lC}W_{jl}^2 \\
&= \sum_{j=1}^{N}\sum_{l=1}^{N} W_{jl}^2 \sum_{C=1}^{\tilde{N}} K_{lC} \\
&= \sum_{j=1}^{N}\sum_{l=1}^{N} W_{jl}^2 \frac{p_l^1}{\sum_{m\in G(l)} p_m^1} \qquad (7.6)
\end{aligned}$$

where $G(l)$ stands for the group to which node l belongs. The inequality was derived from the Jensen's inequality: $\phi(\sum_l a_l x_l) \leq \sum_l a_l \phi(x_l)$, for $\phi(x)$ a convex function (in our case the square function) and $\sum_i a_i = 1$.

We conclude that:

$$||\langle\epsilon^\alpha|WK|| \leq ||\epsilon^\alpha|| \sum_{j=1}^{N}\sum_{l=1}^{N} W_{jl}^2 \frac{p_l^1}{\sum_{m\in G(l)} p_m^1} \leq ||\epsilon^\alpha|| \sum_{j=1}^{N}\sum_{l=1}^{N} W_{jl}^2 = ||\epsilon^\alpha||\,\mathrm{Tr}(WW^T) \qquad (7.7)$$

Eq. (7.7) might seem contradictory since the larger the groups the lower the upper bound. However $||\langle\epsilon|||$ is likely to become larger if each group is made of several nodes. Moreover comparing $||\langle\epsilon^\alpha|WK||$ for different group structures is not always meaningful since the dimension of $\langle\epsilon^\alpha|WK$ changes with the number of groups. A second important remark concerns the interpretation of Eq. (7.5). Since eigenvectors are defined up to a multiplicative factor, we can always assume that $||\langle u^\alpha|K|| = 1$. If λ^α is large (typically close to one), the condition that $||\langle\epsilon^\alpha|WK||$ is small is sufficient to conclude that $\lambda^\alpha\langle u^\alpha|K$ is almost parallel to $\langle u^\alpha|K\tilde{W}$. However, as λ^α becomes much smaller, we might have that $\lambda^\alpha\langle u^\alpha|K$ is not at all parallel to $\langle u^\alpha|K\tilde{W}$ even if $||\langle\epsilon^\alpha|WK||$ is small. Therefore we expect SCG to perform more accurately for large eigenvalues. This has been observed in Section 7.7.2.

These two observations set a limit for the use of $||\langle\epsilon^\alpha|WK||$ as the criterion for evaluating the performance of the coarse graining. In the following we will rather use the quantity $1 - \frac{\langle u^\alpha|K|\tilde{u}^\alpha\rangle}{||\langle u^\alpha|K||\cdot||\tilde{u}^\alpha||}$. However, it is very difficult to show (and maybe not true) that this quantity always scales as ϵ.

Finally we refer the reader to Section 7.8.2 for an other mathematical result showing that $\lambda^\alpha \geq \tilde{\lambda}^\alpha$ for all $\alpha \leq \tilde{N}$, where $\tilde{\lambda}^\alpha$s are eigenvalues of $\tilde{W}$.

7.3.1 Practical implementation

For practical implementations, the most intuitive idea is first to select the S eigenvalues and eigenvectors to be preserved. As stated before, a natural choice is to use the S first non-trivial slow modes. Then I intervals of size $l^\alpha \propto \epsilon$ are defined between the largest and the lowest components along each $\langle u^\alpha|$. Nodes falling in the same interval for each α are grouped together. This procedure is equivalent to a box-covering of the S-dimensional embedding space in which node i is represented as a vector given by the S i^{th} components of $\{\langle u^\alpha|\}_{\alpha=2}^{S+1}$. Boxes (or cells) have a size defined by the l^αs and correspond to S-dimensional rectangular parallelepiped. The

derivation presented in this section ensures that the coarse-grained network has eigenvalues $\tilde{\lambda}^\alpha$, resp. eigenvectors $\langle\tilde{u}^\alpha|$ and $|\tilde{p}^\alpha\rangle$ whose difference with λ^α, resp. $\langle u^\alpha|K$ and $R|p^\alpha\rangle$, is in the order of ϵ. Indeed nothing can be said about the other lower eigenvalues which might change a lot, and for most of them, will simply disappear.

7.3.2 Related works

A related mathematical framework had already been designed by Lafon and Lee [95] considering any partition of a network, and not only the ones satisfying the condition that components of $\langle u^\alpha|$ are close to each other within the groups. Considering a random partition, they have shown that the difference between the projected eigenvectors and the eigenvectors of the reduced network has an upper bound. Strictly speaking, they have proved that $|e^\alpha\rangle$ and $\langle g^\alpha|$ defined as

$$\tilde{W}R|p^\alpha\rangle = \lambda^\alpha R|p^\alpha\rangle + |e^\alpha\rangle \qquad \text{and} \qquad \langle u^\alpha|K\tilde{W} = \lambda^\alpha\langle u^\alpha|K + \langle g^\alpha|, \tag{7.8}$$

satisfy the inequalities:

$$\sum_{C=1}^{\tilde{N}} \frac{(e_C^\alpha)^2}{p_C^1} \leq 2D \qquad \text{and} \qquad \sum_{C=1}^{\tilde{N}} (g_C^\alpha)^2\, p_C^1 \leq 2D$$

with

$$D = \sum_{C=1}^{\tilde{N}} \sum_{i\in C} p_i^1 \sum_{\alpha=0}^{S} (\lambda^\alpha u_i^\alpha - \lambda^\alpha(\langle u^\alpha|K)_C)^2 \tag{7.9}$$

S stands for the number of relevant eigenvalues that have been considered and $(\langle u^\alpha|K)_C$ stands for the C^{th} component of $\langle u^\alpha|K$. One easily checks that if $\tilde{N} = N$, D=0.

The upper bound found in Eq. (7.9) is certainly an interesting feature. However it does not allow to conclude which partition should be chosen so that $\langle u^\alpha|K$ is a good approximation of $\langle\tilde{u}^\alpha|$. A possible way to tackle this problem is to minimize D. The main difficulty is that this minimization is far from trivial. In [95] it has been shown that minimizing D can be addressed by k-means algorithm [100]. However the k-means minimization implies to fix *a priori* the number of groups in the reduced network, which most often cannot be done unambiguously. In comparison, our approach has several advantages. Firstly it ensures that $\langle u^\alpha|R$ is a good approximation of $\langle\tilde{u}^\alpha|$ as long as $l^\alpha \propto \epsilon$ is small. Secondly it only requires to compute the first eigenvectors, without any other post-processing steps. Thirdly, once S and I (the number of intervals along each eigenvectors) have been chosen, the number of nodes in the reduced network is given by the properties of the network itself and not arbitrarily fixed.

Nevertheless the work of Lafon and Lee defines the suitable mathematical framework for coarse graining complex networks. For this reason it deserves a special mention in this Thesis.

Finally in [60] it has been suggested that instead of focusing on the S first eigenvectors, one could compute the distance:

$$d(i,j) = \sum_{\alpha=1}^{N} \left|u_i^\alpha - u_j^\alpha\right| \left|(\lambda^\alpha)^T\right|$$

with $T \in \mathbb{N}$ and group nodes satisfying $d(i,j) < \epsilon$. A series of mathematical results is presented in [60]. The main draw-back of this method is that it requires the computation of all eigenvectors which is time-consuming for large graphs, while the Spectral Coarse Graining presented in this Thesis involves only the first eigenvalues and eigenvectors. Unfortunately this interesting approach has not been applied to real networks and for this reason has been almost completely unnoticed in the field of graph theory and complex networks.

7.3.3 Directed networks

Up to now, we have always assumed undirected networks. Yet directed networks are often encountered when dealing with real networks. For instance gene regulation networks are strongly directed since the regulation process is not symmetric in most occurrences. The World Wide Web is an other instance of a directed network in which the edge direction is crucial for most properties.

To coarse grain directed networks, we consider the strongly connected component (scc) of a network, plus all nodes either pointing to a node of the scc or pointed by a node of the scc. Related to the random walk properties on directed networks, three different scenarii need to be distinguished:

1. There exists a stationary state and each node has a non-zero probability in the stationary state.
2. There exists a stationary state but some nodes have a stationary probability equal to 0.
3. There exists no stationary state.

In the first case, the network is a scc and matrix K is properly defined. The main changes take place because $|p^\alpha\rangle$ can no longer be expressed in terms of $\langle u^\alpha|$ and $|p^1\rangle$ (see the derivation of Eq. (7.2)). Nevertheless the coarse graining along $\langle u^\alpha|$ preserves the eigenvalue λ^α as well the eigenvector $\langle u^\alpha|$. Furthermore because of the particular definition of K, $KR|p^1\rangle = |p^1\rangle$, which ensures that the stationary state $|p^1\rangle$ is always preserved under coarse graining. This result becomes crucial when considering the PageRank matrix [24] for instance. PageRank has received much attention because of its use in search engines like Google. It is defined via a stochastic process on the WWW, where for each step a "random surfer" either follows with probability d one of the existing outgoing links or jumps at random to another site with probability $(1-d)$. PageRank of the nodes corresponds to the stationary state of the process (i.e. to $|p^1\rangle$). Thus under SCG the PageRank in the reduced network is the sum over the PageRank in the initial network. The only effect of the directed nature of the WWW, compared to undirected networks, is that $|p^\alpha\rangle$ is not automatically preserved when coarse graining along $\langle u|^\alpha$ for $\alpha > 1$.

The second case is typically encountered when some nodes cannot be reached from all others. In this case the network is disconnected. Globally disconnected networks are difficult to handle and even the idea of coarse graining is not well-defined, as it was the case with clustering. Nevertheless a few simple cases can be treated. Figure 7.3A and B displays two generic situations in which a node is either unreachable from the others, or act as a trap. From the point of view of a random walk, unreachable nodes do not play any role, and can be removed from the network without altering any spectral property. The presence of traps is more interesting since it relates to the notion of exit probabilities (see Section 7.5). Any trap i is associated with a stationary state $|p^\beta\rangle$ with $\lambda^\beta = 1$ and $p_j^\beta = \delta_{ij}$, which implies that K in Eq. (7.3) is not defined. For a suitable coarse graining approach one should first consider each trap as a single group. Other nodes are then grouped according to their components in a left eigenvector $\langle u^\alpha|$. Since matrix K can be defined in various ways as long as the columns are normalized, a possibility for its non-zero entries is to use the stationary state of a network in which the edges pointing to the traps are symmetric.

The third case reflects the presence of sinks in the network, i.e. nodes acting as absorbing wells. Sinks are especially useful to compute the mean first passage time to a given node [63, 13]. A sink corresponds to a column of the stochastic matrix W set to 0. If the rest of the network is a scc, $|\lambda^\alpha| < 1 \ \forall \alpha$. Yet the coarse graining can still be applied. As in the previous case, a convenient way to define matrix K is to consider the sink node i as a group and to set to one the corresponding non-zero entry in K. Other non-zero elements of K are filled considering the stationary probability over a network in which edges pointing to the sink nodes have been

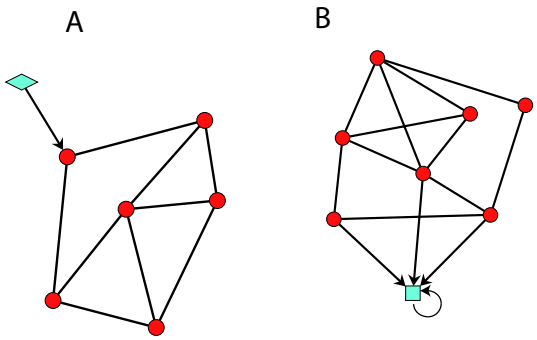


Figure 7.3: Two kinds of networks in which some nodes have a stationary probability equal to 0. **A**: Example of an unreachable node. **B**: example of a trap.

α	λ^α	$\tilde{\lambda}^\alpha$
1	1	1
2	0.3091 + 0.3767i	0.3091 + 0.3767i
3	0.3091 - 0.3767i	0.3091 - 0.3767i
4	-0.5958 + 0.1032i	-0.5958 + 0.1032i
5	-0.5958 - 0.1032i	-0.5958 - 0.1032i
6	-0.4267	-0.4267
7	0	

Table 7.1: Comparison between the spectrum of the two networks of Figure 7.4.

symmetrized. If multiple sinks are present, one can either consider each of them as a single group (for instance if it matters in which node the random walk escapes) or group them into one single sink in the reduced network.

Combining several sinks, traps and unreachable nodes becomes somehow tedious, both from a practical point of view and a theoretical point of view. Since this situation is not often encountered, we will not describe it here.

Finally we note that directed networks are characterized by complex eigenvalues of W. For most real directed networks, it has been observed that complex eigenvalues are not the ones with the largest module and thus do not enter in the Spectral Coarse Graining. However, generally speaking, the module of complex eigenvalues can take any value smaller or equal to one. Complex eigenvalues and eigenvectors always arrive by conjugated pairs, which implies that coarse graining the network along a complex eigenvector is equivalent to coarse graining along its conjugate. Apart from that, the coarse graining can be readily extended to complex eigenvalues by defining intervals in the embedding complex space $\mathbb{C}^S$. In particular the fact that two nodes with exactly the same neighbors have the same eigenvector components for $\lambda^\alpha \neq 0$ still holds, as shown in Figure 7.4 and Table 7.1. An interesting situation occurs when the network contains stable cycles, resulting in complex eigenvalues with a module equal to one. The presence of stable cycles strongly constrains the topology of the network. Figure 7.5 shows an example of such network, which does not trivially consists in one oriented ring. Interestingly, the coarse graining has the effect of merging the different pathways to obtain a ring. We suggest that it might be a general property of directed network with stable cycles to be transformed into rings under SCG. It will be interesting to study in more details these networks and to understand how the almost-exact coarse graining performs.

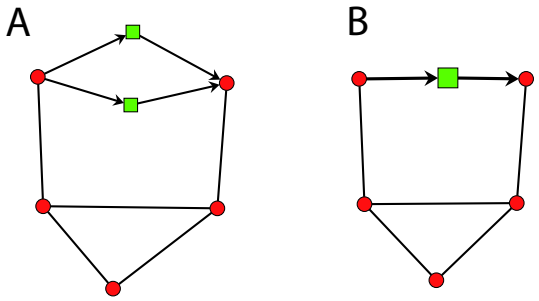


Figure 7.4: Example of a directed network with complex eigenvalues. **A**: Initial network with two equivalent nodes. **B**: Coarse-grained network.

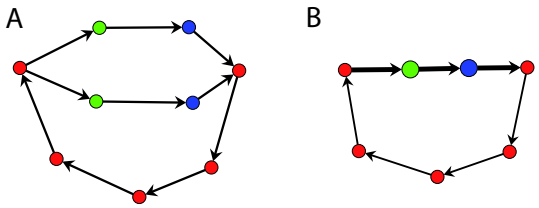


Figure 7.5: Example of a directed network with complex eigenvalues of module 1. **A**: Initial network with two equivalent nodes. **B**: Coarse-grained network.

7.4 Di-alanine network

The concept of random walks is especially appropriate for Configuration Space Networks (CSN) described in Chapter 5. As a quick recall, a node accounts for a configuration sampled during the simulation and edges represent transitions between configurations [148]. In CSNs the weight of an edge represents the number of transitions between two states of the configuration space observed along the simulation and the elements of W correspond to transition probabilities. Therefore random walks on CSNs provide a way to explore the space of configurations without having to run *de novo* the entire simulation.

We applied the coarse graining scheme to the di-alanine network already discussed in Chapter 5 [64]. Detailed balance implies that the network should be undirected, that is $A_{ij} = A_{ji}$, but not $W_{ij} = W_{ji}$. However, because of finite length simulations, a few edges have slightly different weights depending on the direction ($A_{ij} \neq A_{ji}$). In this work the network was considered as undirected by taking the average over the weights in each direction. This pre-processing step does not change the general properties of the network. The network is made of 1832 nodes (Figure 7.6A). In Chapter 5, di-alanine network was shown to consist of four main clusters (colors in Figure 7.6), corresponding to the four main energy basins of the underlying free-energy landscape [64].

To coarse grain the network, we have used the first three non-trivial left eigenvectors $\langle u^2|, \langle u^3|$ and $\langle u^4|$ of W. Along each eigenvector, $I = 60$ intervals of equal size have been defined between the highest and the lowest component. Nodes have been grouped together if they belonged to the same interval along the three eigenvectors. In this way 227 non-empty groups have been found. The coarse-grained network is shown in Figure 7.6B. Colors were set according to the clusters of the nodes in each group. Although the nodes of a group do not necessarily belong to the same cluster, this situation happened only for 4 groups (representing 15 nodes) out the 227. We also applied on the coarse-grained network the same clustering algorithm [47] used to identify the clusters in Figure 7.6A. Exactly 4 clusters were obtained

	α	λ^α	$\tilde{\lambda}^\alpha$	$\frac{\langle u^\alpha \vert K \vert \tilde{u}^\alpha \rangle}{\vert\vert \langle u^\alpha \vert K \vert\vert \cdot \vert\vert \tilde{u}^\alpha \vert\vert}$	$\frac{\langle \tilde{p}^\alpha \vert R \vert p^\alpha \rangle}{\vert\vert R \vert p^\alpha \rangle \vert\vert \cdot \vert\vert \tilde{p}^\alpha \vert\vert}$
A	2	0.99987	0.99987	0.9999	0.9999
	3	0.99947	0.99944	0.9998	0.9999
	4	0.99785	0.99780	0.9999	0.9999
B	2	0.98955	0.98922	0.9985	0.9941
	3	0.98901	0.98861	0.9989	0.9924
	4	0.98779	0.98741	0.9885	0.9686
C	2	0.99971	0.99971	0.999916	0.9999
	3	0.99934	0.99933	0.9994	0.9988
	4	0.99917	0.99916	0.9998	0.9997

Table 7.2: Columns 2 and 3: the three largest (non-trivial) eigenvalues of the stochastic matrices W and $\tilde{W}$. Column 4: Scalar product between $\langle u^\alpha \vert K$ and $\langle \tilde{u}^\alpha \vert$ for the three left eigenvectors used in the coarse graining procedure. Column 5: Scalar product between $R \vert p^\alpha \rangle$ and $\vert \tilde{p}^\alpha \rangle$ for the three right eigenvectors. Box **A**: Di-alanine network shown in Figure 7.6A and B. Box **B**: Erdös-Rényi network. Box **C**: Barabási-Albert network.

corresponding to more than 98% of the initial nodes correctly classified. Thus, even if the aim of Spectral Coarse Graining is different than the usual clustering, the results are indeed consistent with the global features revealed by the cluster structure of the network. In addition the cluster structure is robust under coarse graining.

As expected from the perturbation derivation, the first eigenvalues are preserved in the coarse-grained network with high accuracy (Table 7.2 **A** columns 2 and 3). As for the normalized scalar product, Table 7.2 **A** (columns 4 and 5) shows that the projected left and right eigenvectors $\langle u^\alpha \vert K$ and $R \vert p^\alpha \rangle$ are almost equal the corresponding eigenvectors of $\tilde{W}$. Similar results have been obtained considering the giant component of an Erdös-Rényi network [48] ($N = 5626$, $< k >= 2$, Table 7.2 **B**) and a Barabási-Albert network [11] ($N = 6005$, $m = 1$, Table 7.2 **C**), always considering the three first non-trivial left eigenvectors $\langle u^2 \vert, \langle u^3 \vert$ and $\langle u^4 \vert$ and $I = 60$. The general agreement, though slightly lower for the E-R network, indicates that the perturbation approach is robust for various kinds of networks even if components in $\langle u^\alpha \vert$ are not equal but close to each other within the groups (for a more detailed discussion about robustness of SCG, see Section 7.7).

Figure 7.6 hints that the global architecture of the coarse-grained network is representative of the original one. For instance most nodes buried in the center of the red cluster form one single group, while the nodes lying along the few pathways connecting the red and violet clusters, and therefore critical for the network global connectivity, are well preserved. Interestingly these nodes correspond to the transition states between the four basins. A more stringent test is done by comparing the mean first passage time (MFPT) from node j to node i, T_{ij}. In the context of transport phenomena or search on a network, MFPT is an important characteristic of random walks [133, 13]. To compute it exactly, one usually considers node i as a sink and uses the stochastic matrix $\hat{W}$ with the i^{th} column set to 0 ($T_{ij} = \sum_{t=0}^{\infty} t(\hat{W}^t)_{ij} = \sum_{\alpha=1}^{N} \hat{u}_i^\alpha \hat{p}_j^\alpha \frac{\hat{\lambda}^\alpha}{(1-\hat{\lambda}^\alpha)^2}$). To compare the MFPTs, we used the coarse graining shown in Figure 7.6B, defining the sink node i as a single group. Figure 7.7 shows with black circles ($\circ$) the average MFPT to node i for each group in original network. Two different sinks have been considered (Figure 7.7, A and B). The MFPT to the group representing node i in the coarse-grained network is shown with red lines. The excellent overlap indicates that the MFPT is extremely well preserved, whereas this is not the case in the network of clusters (see onsets in Fig. **??**). Hence the coarse-grained network is representative of the general features of the diffusion process in the initial network. This finding was shown to be robust if other eigenvectors are included, as long as the size of the

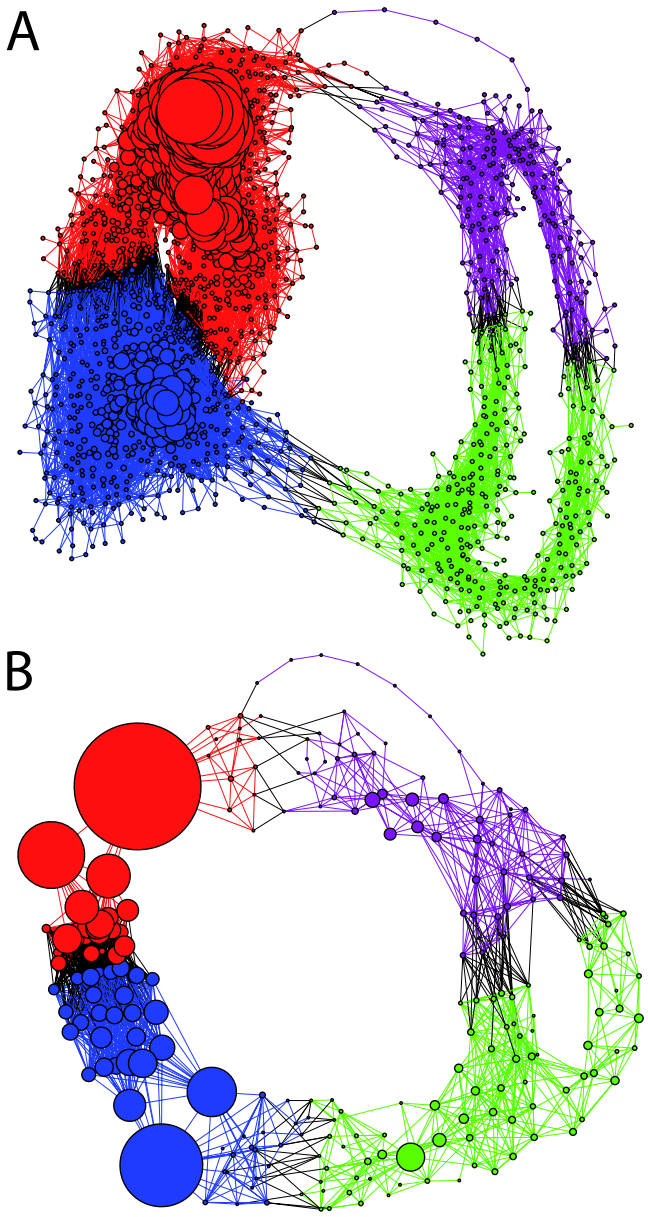


Figure 7.6: **A**: di-alanine network (N=1832). Node size is proportional to their weight (i.e. the number of times nodes have been visited in the simulation). The four different colors correspond to the clusters found in Chapter 5. **B**: Coarse-grained network ($\tilde{N} = 227$) according to $\langle u^\alpha|$, $\alpha = 2, 3, 4$ and $I = 60$. Node size is proportional to the total weight of the groups.

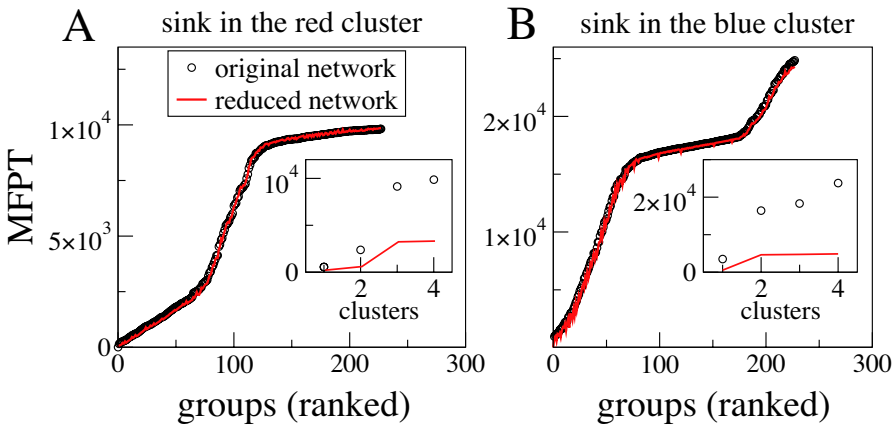


Figure 7.7: Ranking of the MFPT. The circles (○) represent the average MFPT for each group in the original network (variances are not shown since they are always smaller than the size of the circles). The MFPT of the corresponding nodes in the coarse-grained network is displayed with red lines. **A**: di-alanine network with the sink i as the heaviest node of the red cluster . **B**: di-alanine network with the sink i as the heaviest node of the blue cluster. Onsets: Comparison of the MFPT between the original network (○) and the network of clusters (red line).

intervals is kept small enough. In this respect the value $I \propto \epsilon^{-1}$ tunes the degree of precision: increasing I improves the agreement between the initial and the coarse-grained network, but in the same time results in a larger $\tilde{N}$.

Another interesting feature of random walks is the evolution of the probability $P(j,t|i,0)$ of being in node j at time t having started in node i at time 0. In the coarse-grained network, we compute the quantity $\tilde{P}(B,t|A,0)$, where A is the group of node i and B the group of node j. If the coarse graining is exact, $P(j,t|i,0)$ should be equal to $\tilde{P}(B,t|A,0)\cdot P(j,\infty)\cdot\tilde{P}^{-1}(B,\infty)$. Figure 7.8 shows that the coarse-grained network of Figure 7.6 exhibits almost the same time evolution, whereas the network of cluster yields significant discrepancies. This observation illustrates well the differences between coarse graining and clustering. In the clustering approach, groups correspond to large-scale features of the network, but random walks on the network of clusters are not equivalent to those on the initial network. On the other hand groups in the coarse-grained network do not account for global feature of the network, but the coarse-grained network displays the same dynamical behavior as the initial one.

7.5 Exit probabilities

In general the largest eigenvalues and eigenvectors of W represent the large scale behavior of random walks. However, in some cases eigenvectors are directly associated with useful quantities, such as PageRank discussed in Section 7.3.3. As a different example we consider a random walk on a network with two traps (say node 1 and N). The stochastic matrix reads:

$$W_{exit} = \begin{pmatrix} 1 & W_{12} & \dots & 0 \\ 0 & W_{22} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & W_{N2} & \dots & 1 \end{pmatrix}$$

W_{exit} has two eigenvalues equal to one, corresponding to an eigenspace of dimension 2. Writing $W = \sum_{\alpha=1}^{N} \lambda^\alpha |p^\alpha\rangle\langle u^\alpha|$ with $\langle u^\alpha|p^\beta\rangle = \delta_{\alpha\beta}$ and taking $|p^1\rangle = (1,0,...,0)$ and $|p^2\rangle = (0,...,0,1)$

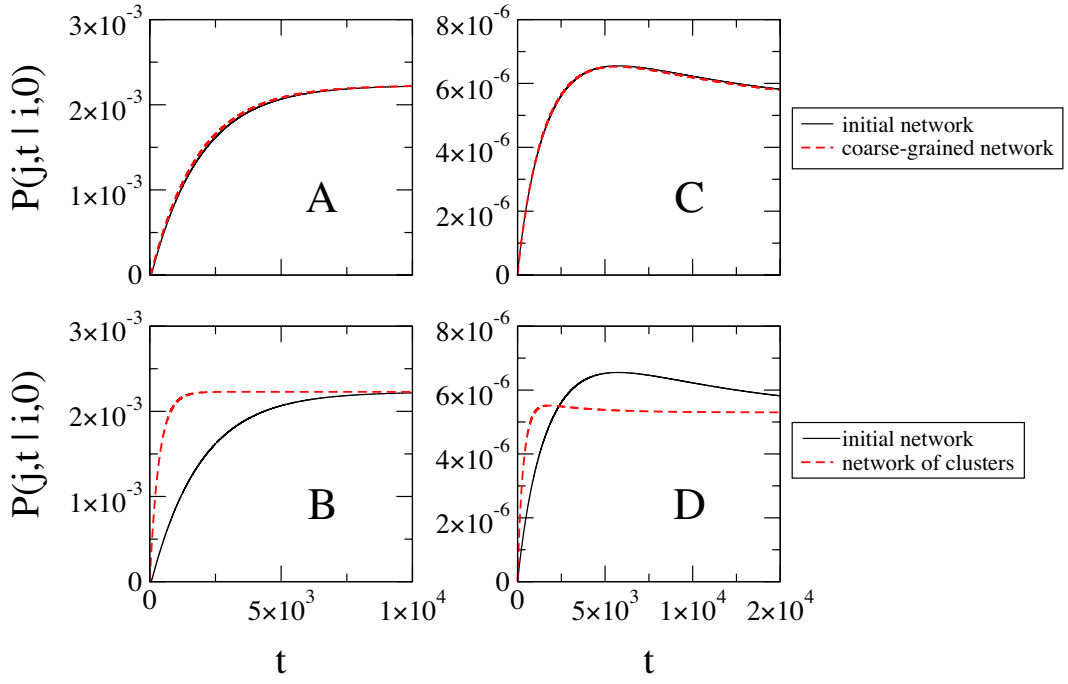


Figure 7.8: Time evolution of random walks on di-alanine network. Continuous black line shows $P(j,t|i,0)$. Dashed red lines display $\tilde{P}(B,t|A,0)\cdot P(j,\infty)\cdot\tilde{P}^{-1}(B,\infty)$. A is the group of node i and B the group of node j. **A** and **B**: i is chosen as the central node of the red cluster and j as the central node of the blue cluster in Figure 7.6. **C** and **D**: i is chosen as a node of the green cluster and j a node of the violet cluster in Figure 7.6. **A** and **C**: groups are defined with the coarse graining of Figure 7.6B. **B** and **D**: groups are defined as the 4 clusters found by MCL.

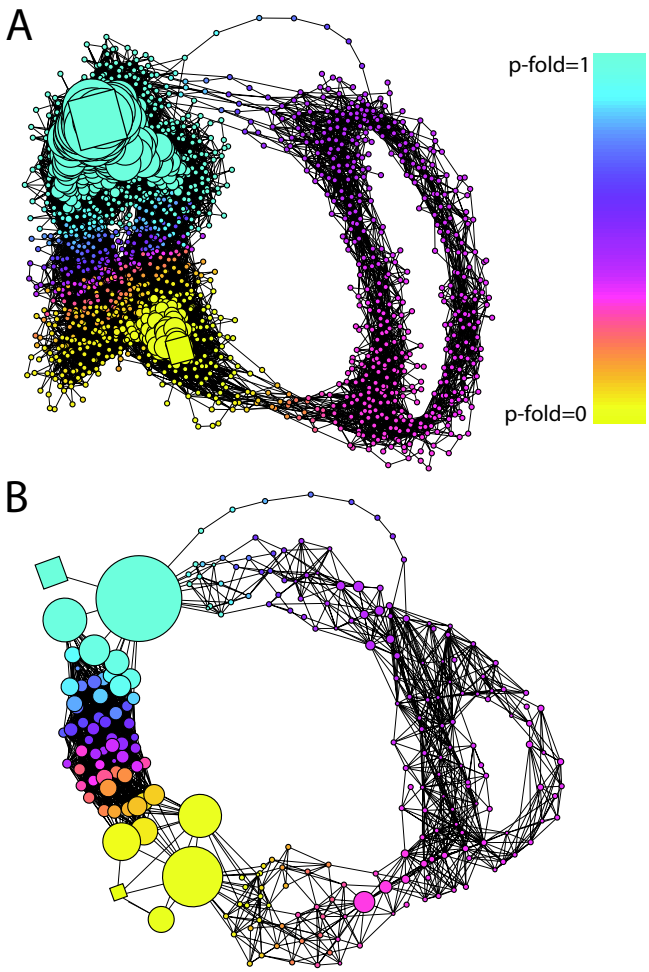


Figure 7.9: Coarse graining along the p-fold. **A**: Original di-alanine network. **B**: Coarse-grained di-alanine network along the left eigenvectors $\langle u^1|$, $\langle u^3|$ and $\langle u^4|$ of W_{exit} (I=60). The color map reflects the exit probability. Square nodes represent the two traps.

the probability of exit in node 1 starting at node i is given by:

$$P_j^{out}(1) = \lim_{t\to\infty}(W_{exit}^t)_{1j} = u_j^1 \qquad j = 2, ..., N-1$$

Thus $\langle u^1|$, resp. $\langle u^2| = \langle 1| - \langle u^1|$, gives the probability of exit in node 1, resp. node N.

In the case of di-alanine network, the natural choice for nodes 1 and N is the most populated nodes of the two main basins, since they act as representatives of the native and denaturated states. With this choice the exit probability is associated with the p-fold [43], defined as the probability to reach the native state before visiting the denaturated state[2]. P-fold has been used as an order parameter on which the high-dimensional energy landscape is projected [43].

In Figure 7.9 the coarse graining of di-alanine network was performed using $\langle u^1|$, $\langle u^3|$ and $\langle u^4|$ (one could equivalently use $\langle u^2|$ instead of $\langle u^1|$). The two square nodes represent the two traps. Nodes color correspond to the exit probability. coarse graining along $\langle u^1|$ ensures to preserve the exit probability (see Figure 7.10), while including $\langle u^3|$ and $\langle u^4|$ allows to preserve the general structure of the network.

[2]Note that if the denaturated state is not known or consists in a large ensemble of states, p-fold is defined in a slightly different way [82].

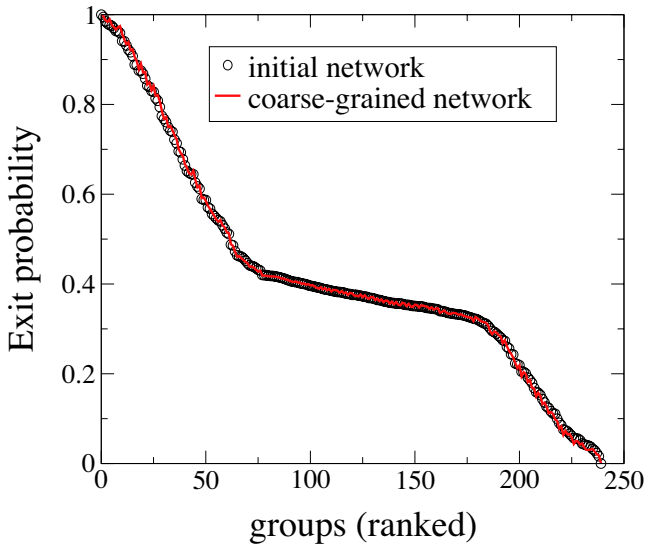


Figure 7.10: Exit probability in the original and reduced di-alanine networks. Groups have been ranked according to components in $\langle u^1|$.

Therefore our method allows to coarse grain a network in such a way that the p-fold is almost perfectly preserved for every configuration. It can be further reinterpreted as a projection of the network onto the order parameters defined by p-fold and some other eigenvectors.

Interestingly a similar attempt can be found in a work of Rhee *et al.* [154]. Considering a D-dimensional energy landscape with two sinks at two different locations, the authors have designed a continuous mapping onto a one dimensional potential such that the mapping preserves the exit probability. Their work can be regarded as a continuous version of SCG along the first left eigenvector of W_{exit}.

Finally, our results suggest that considering only the exit probability is not enough to approximate correctly the folding process. For instance in Figure 7.9 nodes with exit probability equal to 0.5 are located both in the region between the two large clusters, and on the pathways on the right. If the coarse graining had been done only along $\langle u^1|$, such nodes would have been merged and the overall structure would have changed. Thus including a few other eigenvectors allows both to preserve the p-fold and the large scale structure of the network. In this sense the framework of CSNs appears once again as very convenient to analyze protein dynamics. Approximations that were commonly done, such as projecting the dynamics on a few order parameters, can be formalized in the network approach. Furthermore order parameters arise naturally as the first eigenvectors of the stochastic matrix (or equivalently the principal components), which allows to avoid arbitrary choices.

7.6 Cell-cycle networks

Not all networks are characterized by a dynamics described in terms of random walks. However, it has been shown that random walks often can be used as a tool to extract information about the network. For instance several clustering algorithms are based on the idea of performing random walks on networks [172, 96]. In this section we apply the coarse graining scheme on a cell cycle network recently studied in [29]. Though edges are not associated with transition probabilities as in the di-alanine network, SCG based on random walks provides several interesting insights in the network. In particular it shows that the use of SCG is not restricted to complex networks

	α	λ^α	$\tilde{\lambda}^\alpha$	$\frac{\langle u^\alpha \vert K \vert \tilde{u}^\alpha \rangle}{\vert\vert \langle u^\alpha \vert K \vert\vert \cdot \vert\vert \tilde{u}^\alpha \vert\vert}$	$\frac{\langle \tilde{p}^\alpha \vert R \vert p^\alpha \rangle}{\vert\vert R \vert p^\alpha \rangle \vert\vert \cdot \vert\vert \tilde{p}^\alpha \vert\vert}$
1	2	0.98195	0.98173	0.9999	0.9999
	3	0.88389	0.88033	0.9998	0.9995
2	2	0.9794	0.9793	0.9999	1
	3	0.9123	0.9114	0.9999	0.9997
3	2	0.9674	0.9671	0.9999	0.9999
	3	0.8912	0.8893	0.9999	0.9979

Table 7.3: Columns 2 and 3: the two largest (non-trivial) eigenvalues of the stochastic matrices W and $\tilde{W}$. Column 4: Scalar product between $\langle u^\alpha | K$ and $\langle \tilde{u}^\alpha |$ for the two left eigenvectors used in the coarse graining procedure. Column 5: Scalar product between $R|p^\alpha\rangle$ and $|\tilde{p}^\alpha\rangle$ for the two right eigenvectors. Box **1**: Data from Oliva *et al.* [135]. Box **2**: Data from Peng *et al.* [142]. Box **3**: Data from Rustici *et al.* [155].

built from dynamical processes.

Cells are known to exhibit a chronological cycle of events, cumulating in cell division. The cycle can be divided into two main phases. First the doubling of the genome (phase S) takes place. Later on the genome is halved during mitosis (phase M). The period between phase S and M is referred to as phase G1, and the one between M and S as G2. The biochemical origin of this cycle can be found in several genes that display periodic expression profiles. To identify cell cycle genes from their expression profile, micro-array technologies have been extensively used and data have been interpreted with the help of Fourier analysis [32]. As a result a set of genes have been identified as periodically expressed along the cell cycle in several organisms. Since most periodic genes have the same frequency (corresponding to the lifetime of the cell), the key parameter to distinguish between them, and hopefully recover the cell cycle dynamics, is the phase of each gene. Such data are available for the yeast fission *Schizosaccharomices pombe* in [155, 142, 135].

A recent study [29] has shown that the framework of complex networks is convenient to represent cell cycle genes. The network is defined considering each gene as a node. The weight over the edges represents the phase difference and was set to $w_{ij} = \exp\{\beta \cos(\phi_i - \phi_j)\}$, with $\beta = 10$. This definition ensures that the network is undirected, though complete, and that genes with similar phases are strongly connected to each other. To visualize the network, a threshold can be defined on the edges weight. Removing all edges corresponding to a phase difference $|\phi_i - \phi_j| > 0.2$, the network exhibits a circular structure [29]. Figure 7.11A shows the networks extracted from the data of [155], [142] and [135]. Colors account for the cell cycle phase in which each gene is known to take part.

In [29] the community structure of this network has been studied in details using MCL [172] and the stability analysis described in Chapter 4. In particular the clustering entropy has been extensively used as a criterion to find the most stable partition. MCL combined with the clustering entropy allowed to unveil the internal structure of the communities that resembles the one defined by the 4 phases of the cell cycle.

We applied SCG to the three networks without any threshold. Eigenvectors $\langle u^2|$ and $\langle u^3|$ with $I = 30$ have been considered in the coarse graining. As for the di-alanine network, eigenvalues and eigenvectors are preserved with an excellent accuracy, albeit the small number of intervals (Table 7.3).

Once groups have been identified, the coarse-grained network can be visualized considering only edges with a weight above the threshold used in Figure 7.11A. Results are shown in Figure 7.11B. The circular shape is well preserved, and most nodes within a group are part of the same phase. Several other interesting features emerge from the coarse-grained network.

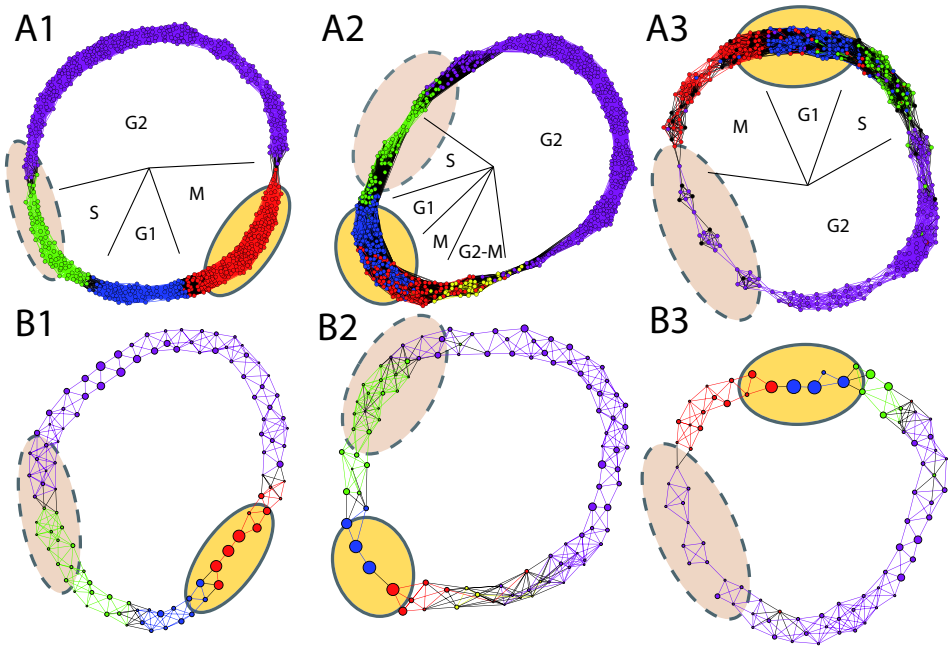


Figure 7.11: **A**: Three cell cycle networks built from the data of Oliva *et al.* [135] (A1), Peng *et al.* [142] (A2) and Rustici *et al.* [155] (A3). Only genes with a phase difference lower than 0.2 have been connected for the visualization. Colors represent the different cell cycle phases. **B**: Coarse-grained cell cycle networks.

From the point of view of random walks, the dynamics on the coarse-grained network agrees with the initial one. Figure 7.12 shows the evolution of the probability $P(j,t|i,0)$ of being in a given node j starting from a node i. The site i was randomly chosen and the node j was taken as a gene strongly correlated to i in Figure 7.12A and weakly correlated to i in Figure 7.12B. As in Figure 7.8 the probability $P(j,t|i,0)$ is compared to $\tilde{P}(B,t|A,0)\cdot P(j,\infty)\cdot \tilde{P}^{-1}(B,\infty)$, with A the group of node i and B the group of node j. In both cases and for all three networks, the temporal dynamics shows that the coarse-grained network behaves exactly as the initial one.

Another interesting feature appears when considering the regions separating the G2 phase from the other phases S or M (dashed ovals in Figure 7.11). Those regions are well preserved and most groups in the coarse-grained network contain only a few nodes. It has been observed that they act as bottlenecks in the network [29]. The corresponding genes are mostly expressed during the main two transitions along the cell cycle and have been suggested as potential new cell cycle regulators. Because of their particular roles, such genes are crucial to be preserved in any coarse graining approach aiming at reducing the complexity of the total cell cycle. On the other hand regions of high density such as the M phase (continuous ovals in Figure 7.11) correspond to very homogeneous regions and might be strongly coarse-grained without modifying the important features of the network. Spectral Coarse Graining presented here is therefore a good candidate to simplify networks of periodic genes, while preserving the most important features of the cell cycle.

7.7 Discussions

An essential issue to deal with is the sensitivity of the coarse graining to the choice of the parameters. The previous examples indicate that in several different kinds of networks, the

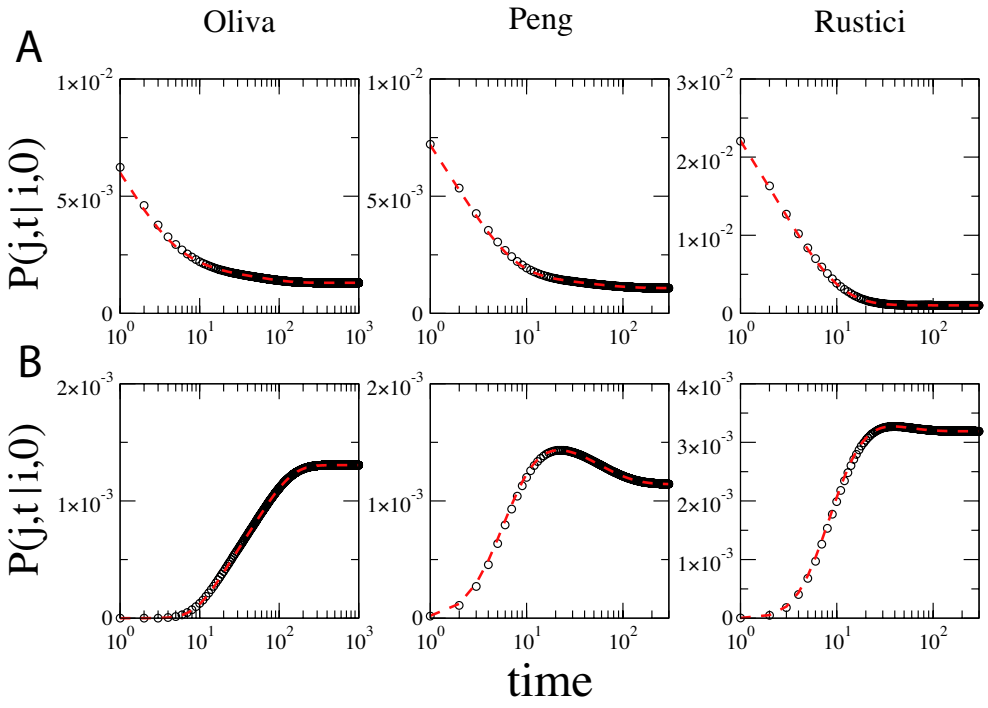


Figure 7.12: Time evolution of the probability $P(j,t|i,0)$ of being in node j having started in i. Circles ($\circ$) display results for the initial networks. Dashed red lines account for the coarse-grained networks. **A**: Node i and j are strongly correlated. **B**: Node i and j are weakly correlated.

coarse graining is robust and intervals along each eigenvector do not need to be very small in order to preserve the spectral properties of the network. In this section a more general test is carried out and a few important features emerge about the choice of parameters S and I, as well as about the kinds of networks particularly suited for a coarse graining approach.

7.7.1 Choosing the parameters

Two parameters have been used in SCG: the number of eigenvectors S and the number of intervals I along each eigenvector. These two parameters have different roles. Since the information about the network (or about W) is spread among the eigenvectors with eigenvalues $\lambda \neq 0$, S represents the amount of large scale information that we want to preserve from the original network. Referring to Eq. (7.1), $\lambda^1, ..., \lambda^{S+1}$ can be regarded time scales that will be preserved in the coarse graining.

In some situations the choice of S is based on a well-defined criterion. This is the case if a significant gap is present between a few eigenvalues close to one and the rest of the spectrum. For networks consisting of a few number of nodes (typically 50-100) and exhibiting a clear community structure, the gap can be identified relatively easily. However when dealing with larger networks characterized by a fuzzy community structure, the gap turns out to be extremely delicate to identify unambiguously (see Figure 7.13). Thus looking for the gap can not be considered as a general criterion, at least when dealing with real networks. An alternative is to consider eigenvalues λ^{α^t} of the matrix power W^t [60]. Even if no significant gap is visible in the spectrum of W, taking the t^{th} power will increase the relative differences between eigenvalues. For a large t some eigenvalues might still scale $\mathcal{O}(1)$, while all others may be several order of magnitudes smaller. In this way, the choice of the "relevant" eigenvectors is given by the time

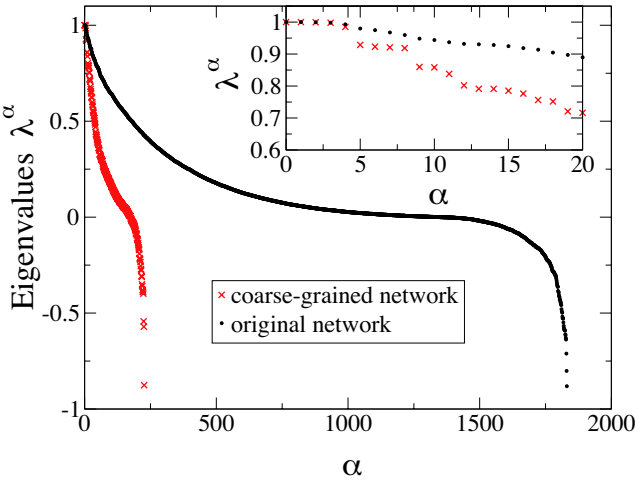


Figure 7.13: Ordered list of di-alanine network eigenvalues. Onset: zoom on the largest eigenvalues.

scale t and the spatial effect of SCG (i.e. grouping nodes) is immediately related with a temporal coarse graining. Nonetheless, even if a global criterion for the choice of S is often difficult to find, the examples presented above show that reasonable results are often obtained, even if S was somehow chosen arbitrarily.

Parameter I gives the accuracy with which this information is preserved. The larger I, the better the precision. In the ideal case ($I = \infty$), we have shown that all relevant eigenvalues are exactly preserved. Moreover we have seen that only eigenvalues equal to 0 (see section 7.2) or pathological eigenvalues (see Appendix B.1) are removed from the spectrum. Thus no information is lost and the coarse graining is exact. Setting I to a finite value induces some changes. First of all, slight differences between the S first eigenvalues might appear, though the perturbation derivation shows that these differences are small if the number of intervals is large enough. Concerning other eigenvalues, nothing can be said *a priori*, except that most of them will not be preserved since the number of nodes of the coarse-grained network is significantly smaller than the number of non-zero eigenvalues of W. Figure 7.13 shows for instance the spectrum of di-alanine network before and after coarse graining. As it was already pointed out, the first 4 eigenvalues are well-preserved. The fifth one of the coarse-grained network is still close to the one of the initial network. From the sixth one, significant discrepancies are observed.

A useful issue is to find how I should be chosen, and even more generally how nodes should be grouped, so that the number of groups is minimum, but the precision in the eigenvalues is still good. Several schemes can be designed to find an optimal coarse graining. But before exploring some of them, one important thing has to be stressed. Since we are coarse graining and not clustering a network, the group in which a node falls is not crucial, as long as the coarse-grained network is representative of the initial one. Thus the motivation to find an optimal way of grouping nodes arises mostly from a trade-off between size reduction and the precision of SCG.

In a first attempt, we can fix a threshold on the precision of eigenvalues and then look for the minimal number of intervals satisfying the condition induced by the threshold. This method requires to compute the eigenvalues of the reduced networks for several values of I. However, since the precision is most often a monotonic function of I, at least for networks on which the coarse graining is meaningful, we can use fast converging algorithms as Newton's method. Figure 7.14 shows the di-alanine coarse-grained network for three different thresholds (I has been chosen minimal such that the threshold was still satisfied). Instead of fixing a threshold

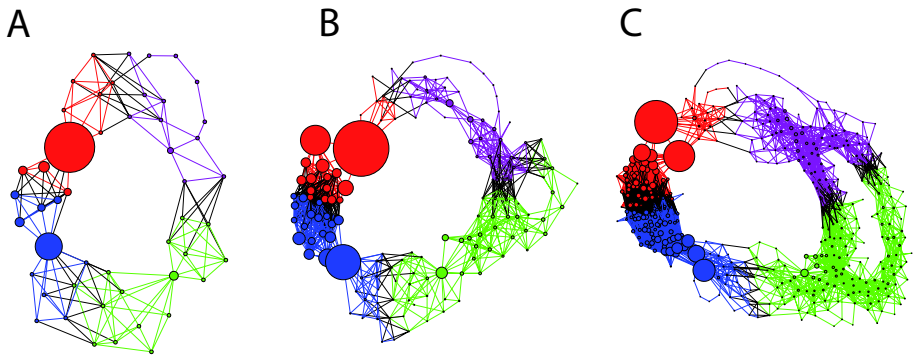


Figure 7.14: Coarse graining with a threshold on the precision $(\lambda^\alpha - \tilde{\lambda}^\alpha)/\lambda^\alpha > \theta$ for $\alpha = 2, 3, 4$. **A**: $\theta = 10^{-3}$, **B**: $\theta = 10^{-4}$, **C**: $\theta = 10^{-5}$

on the eigenvalues, we also investigated the effect of a threshold on the norm $\|\langle\epsilon|\|$, with $\langle\epsilon|$ being equal to $\langle u^\alpha|KR - \langle\tilde{u}^\alpha|$ as in Eq. (7.4).

Alternatively, one can define cells of different shapes in the embedding S-dimensional space of eigenvector components. A very intuitive idea is to give more importance to the largest eigenvalues. This can be done by defining smaller intervals along the eigenvectors with the largest eigenvalues, and larger intervals along those with smaller eigenvalues.

The position and the size of intervals may also be varied to take into account the spatial distribution of the nodes in the embedding space. In this way it is possible deal with the arbitrary position of the intervals along the eigenvectors. From a general point of view, the main idea is to combine a minimization of the number of groups with a minimization of the distances within each group. In this respect we propose to maximize the following measure:

$$Q_{c-g} = \sum_C \sum_{i<j\in C} \exp\left\{\left(\sum_{\alpha=2}^{S+1} \frac{(u_i^\alpha - u_j^\alpha)^2}{(u_{\max}^\alpha - u_{\min}^\alpha)^2}\right)^{1/2}\right\} + \gamma \sum_C \frac{n_C(n_C-1)}{2} \tag{7.10}$$

where $u_{\max}^\alpha$, resp. $u_{\min}^\alpha$, is the maximal, resp. minimal, value along $\langle u^\alpha|$, and n_C stands for the number of nodes in group C. The first term is maximized if groups consist in one node. The second term is maximized if all nodes form a single group. As in [149], parameter γ allows to tune the average distance between the nodes of a group since $\sum_{i<j\in C} \exp\left\{\left(\sum_{\alpha=2}^{S+1} \frac{(u_i^\alpha - u_j^\alpha)^2}{(u_{\max}^\alpha - u_{\min}^\alpha)^2}\right)^{1/2}\right\}$ is equal to $< d_C > \frac{n_C(n_C-1)}{2}$ with $< d_C >$ the average of $\exp\left\{\left(\sum_{\alpha=2}^{S+1} \frac{(u_i^\alpha - u_j^\alpha)^2}{(u_{\max}^\alpha - u_{\min}^\alpha)^2}\right)^{1/2}\right\}$ over all pairs of nodes in group C.

Several ways can be designed to maximize Q_{c-g}. In order to find the global maximum, simulated annealing may be the most reliable approach. However, this method is time consuming and will not be suited for large networks. A steepest descent algorithm can also be designed in which nodes are successively merged. The choice of which nodes (or groups) to merge is the one that results in the largest increase in Q_{c-g}. Results are displayed in Fig 7.15 for an ER graph with average degree $< k >= 2$ and for di-alanine network. The continuous lines show the precision $\frac{\lambda^\alpha - \tilde{\lambda}^\alpha}{\lambda^\alpha}$ $(\alpha = 2, 3, 4)$ for a coarse graining varying the number of intervals I. The dashed lines show the results of Q_{c-g} optimization for various γ. For the sake of comparison the relative difference between $\tilde{\lambda}^\alpha$ and λ^α is displayed as a function of the number of nodes in the coarse-grained network. As it can be seen, the agreement between $\tilde{\lambda}^\alpha$ and λ^α is slightly better when optimizing Q_{c-g}.

In general optimizing the grouping in the embedding S-dimensional space of a network is similar to spatial clustering. However in the coarse graining the condition that nodes are close

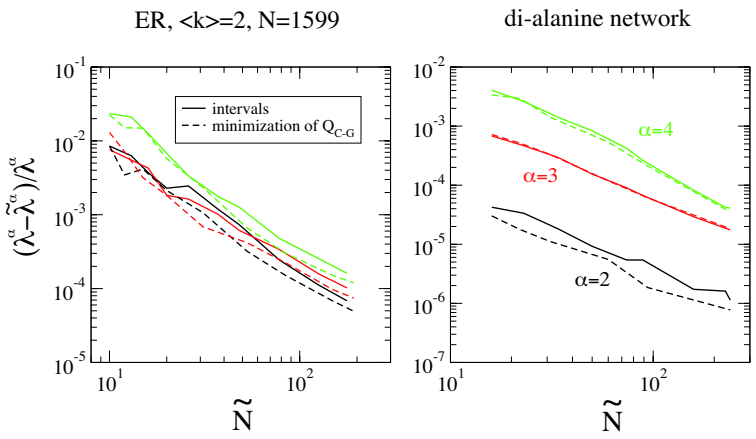


Figure 7.15: Comparison between the coarse graining with I intervals (continuous lines) and the optimization of Q_{c-g} (dashed lines). The coarse graining was always performed on the three left eigenvectors $\langle u^2|$, $\langle u^3|$, $\langle u^4|$. To allow for comparison, relative differences have been displayed as a function of the number of groups $\tilde{N}$, corresponding to different choices of I, resp. γ.

to each other within each group should be strictly satisfied, whereas in the spatial clustering problem, nodes with a given (sometimes large) distance may be grouped together if all their neighbors are much further apart from them. This exemplifies one of the major differences between clustering and coarse graining and provides the main justification for the exponential function in Eq. (7.10).

To summarize, several different schemes can be defined to find an optimal coarse graining. Very advanced techniques might be used for relatively small networks on which running an algorithm scaling as N^2, N^3 or even worse is not a problem. For larger networks, computational time sets a limit. Our initial approach of grouping nodes by discretizing the space into cells of the same size has the advantage both of simplicity and time efficiency, since it requires to compute only a few eigenvectors. It was shown to perform extremely well on several examples of complex networks.

7.7.2 Can any kind of networks be coarse-grained?

Complex networks form an extremely heterogeneous class of mathematical objects, ranging from random graphs [48] to scale-free networks [11] or ordered lattices. For this reason any method may perform well on some kinds of networks, and not on others. We have already encountered such an example when considering clustering techniques. On the one hand if a network consists of clear communities in which nodes are well connected, running a clustering algorithm allows to extract meaningful informations about the network. On the other hand clustering random graphs does not provide any meaningful information and turns out to be highly sensitive to noise.

Within the coarse graining framework, the goals of the procedure are well defined, contrary to the paradigm of clustering. Since we aim at preserving the first eigenvalues and eigenvectors, we can naturally test how sensitive the method is with respect to I for various kinds of networks. Figure 7.16A shows how close $\tilde{\lambda}$ is to λ for $\alpha = 2, 3, 4$ in three different random graphs. Only the giant component has been considered with a size $N = 5626$, resp. $N = 4923$ and $N = 4988$, for $< k >= 2$, resp. $< k >= 4$ and $< k >= 6$. The coarse graining was always performed along $\langle u^2|, \langle u^3|, \langle u^4|$. As expected, the larger I, the better the precision. In Figure 7.16A the loss of precision of the coarse graining corresponds to a decrease in the absolute value of λ^α. The same

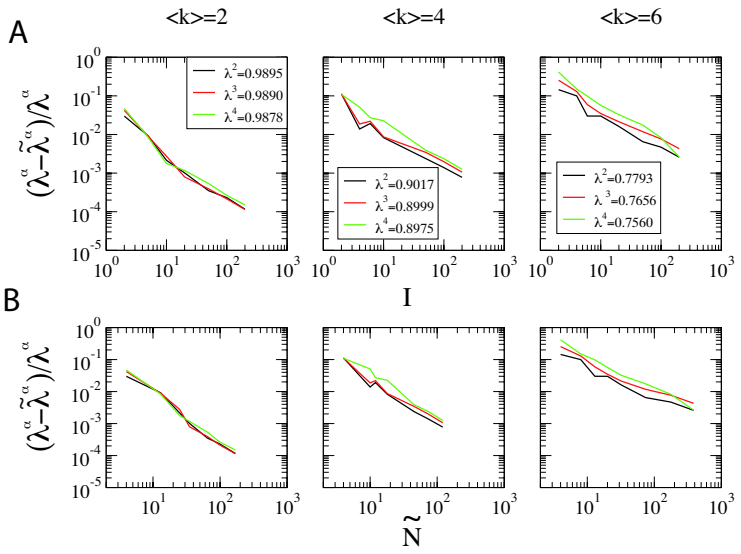


Figure 7.16: Relative difference between λ^α and $\tilde{\lambda}^\alpha$ for the giant component of three random graphs characterized by $< k >= 2, 4, 6$. **A**: $\frac{\lambda^\alpha - \tilde{\lambda}^\alpha}{\lambda^\alpha}$ as a function of I. **B**: $\frac{\lambda^\alpha - \tilde{\lambda}^\alpha}{\lambda^\alpha}$ as a function of $\tilde{N}$.

loss of precision is observed if we use the number of groups instead of the number of intervals on the x-axis (Figure 7.16B).

This behavior is not only characteristic of random graphs and was also observed in Barabási-Albert networks (Figure 7.17). As the number of connections (m) a node makes when entering the network increases, the eigenvalues take smaller values and the precision of SCG decreases.

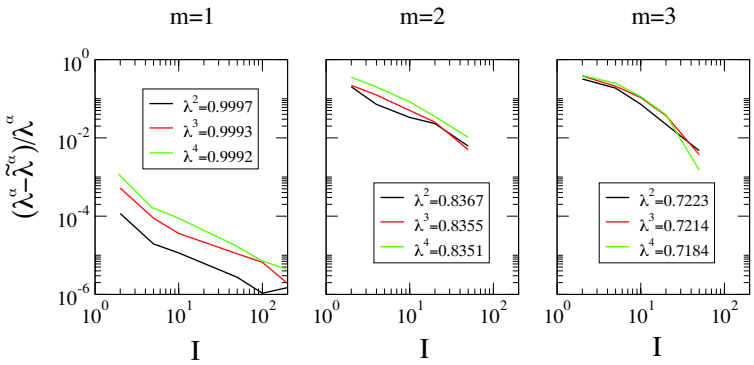


Figure 7.17: Relative difference between λ and $\tilde{\lambda}$ for Barabási-Albert networks built with $m = 1, 2, 3$; $N = 6005$.

This finding was also suggested in [95], since the approximation of λ by $\tilde{\lambda}$ becomes more relevant if $\lambda >> \sqrt{D}$ (see Eq. (7.8) and (7.9)). Thus large eigenvalues allow for a better coarse graining. This is typically the case if the diameter of the network is large, or if the network has an internal structure such as the presence of several communities. These kinds of networks are natural candidates for a coarse graining approach. On the other hand networks characterized by a short diameter and no community structure are likely to perform poorly under any coarse graining approach, due to the absence of any internal structure.

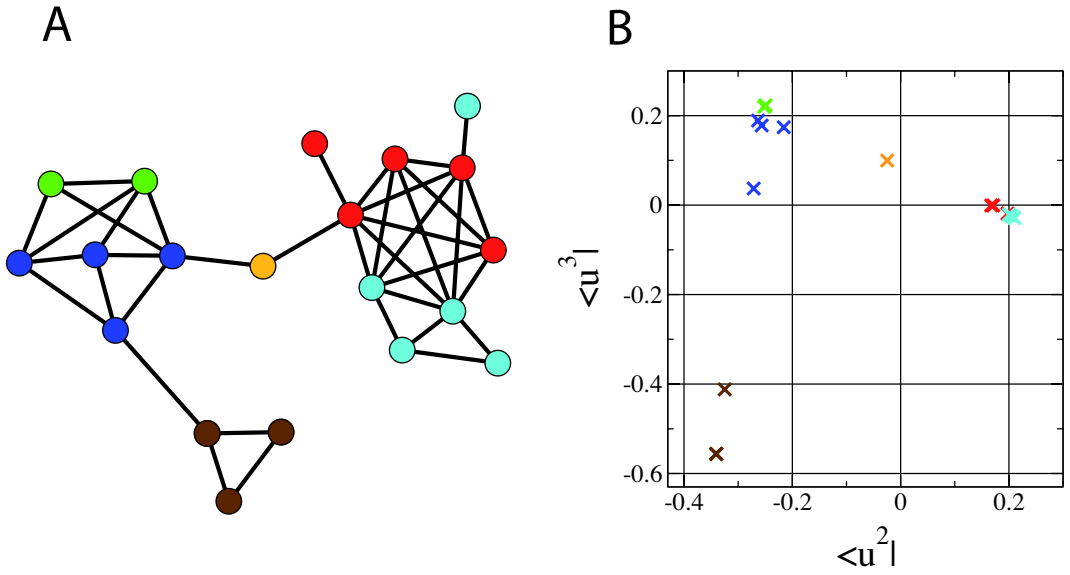


Figure 7.18: SCG along $\langle u^2|$ and $\langle u^3|$, resulting in a pathological clustering. **A**: Small networks with clear clusters that are not recovered in SCG with arbitrary positioned intervals (colors represent the groups found with SCG). **B**: 2 dimensional embedding space of the network. Each node is represented by the coordinate (u_i^2, u_i^3).

7.7.3 Connection with clustering techniques

The ultimate goal of a coarse graining procedure is clearly different than the one followed by most clustering approaches. Nevertheless the techniques underlying the two procedures are related, since both approaches are based on the idea of grouping nodes. Here we will discuss the connection between Spectral Coarse Graining and spectral clustering.

Spectral clustering of networks is based on the eigenvectors of either the Laplacian or the normal matrix [28, 38], which is the transpose of the stochastic matrix W. Remarkably, it is well known that if some nodes belong to the same cluster they tend to have correlated components in the S first non-trivial left eigenvectors of W (see also Section 3.2.7). However for practical implementations, a grouping restricted to nodes whose components are very close to each other (i.e. falling in the same box of length scale $l^\alpha \propto \epsilon$ in the coarse graining framework) performs poorly and the "correct" clusters might be split into several parts, as shown in Figure 7.18. Even if clusters appear as well-defined groups in the left eigenvector components, it may happen, because of the arbitrary position of intervals, that the mean value of eigenvector components within a cluster falls exactly on the interface between two cells of the embedding space (see Figure 7.18). Although this does not alter the validity of the coarse graining (eigenvalues and eigenvectors are always preserved), it is certainly not reasonable from a clustering perspective.

A possible way around this problem is to relax the grouping. We propose the following strategy as a balance between coarse graining and clustering, such that the spectral properties of a network are still preserved, but groups (at least in the case of a clear community structure) can be interpreted as clusters. Following the coarse graining approach, I intervals of equal size l^α are defined between the largest and the smallest components of each $\langle u^\alpha|$, $\alpha = 2, ..., S+1$. As before nodes are grouped together if they belong to the same interval for all α. Then two groups are merged if the difference between the average of the components of $\langle u^\alpha|$ in each group is smaller than the interval size l^α for all $\alpha = 2, ..., S+1$. In this way we can deal with the arbitrary aspect of the position of the intervals and still have the property that the nodes

i	λ	$\tilde{\lambda}$
2	0.997454	0.997358
3	0.995172	0.994854
4	0.984384	0.984193
5	0.98268	0.982145

Table 7.4: Four non-trivial largest eigenvalues of the stochastic matrix for the two networks shown in Figure 7.19A and B.

of a group have strongly correlated components along each left eigenvector. For instance in Figure 7.18 the red and cyan nodes would form one single group, as well as the green and blue nodes, after relaxing the coarse graining.

We applied this approach to a small network built from a synonymy relation between words [61] (see Section 4.5, Figure 4.6). It consists of 185 nodes and is displayed in Figure 7.19A. Figure 7.19B shows the result of the coarse graining over the first 4 non-trivial eigenvectors ($I = 20$) including the relaxation described above. From a clustering point of view the different groups identified in this way correspond to reasonable clusters. From a coarse graining point of view the spectral properties are preserved, as shown in Figure 7.19C and D and Table 7.4. Figure 7.19 allows to conclude that for networks characterized by a clear community structure, the coarse graining approach can be extended to work as a clustering algorithm that preserves the spectral properties of the network.

However, after testing the method over different networks with fuzzier clusters than in Figure 7.19, it was clearly outperformed by the existing clustering algorithms, such as those optimizing the modularity [75, 38, 33, 150]. Thus identifying the correct clusters as aimed by clustering technique and preserving the spectral properties with good accuracy is not always possible for real networks characterized by a fuzzy community structure. Given the nature of the problem, the two approaches (clustering and coarse graining) might be equally relevant, each unveiling different features of the network organization as it was shown in the case of the di-alanine network.

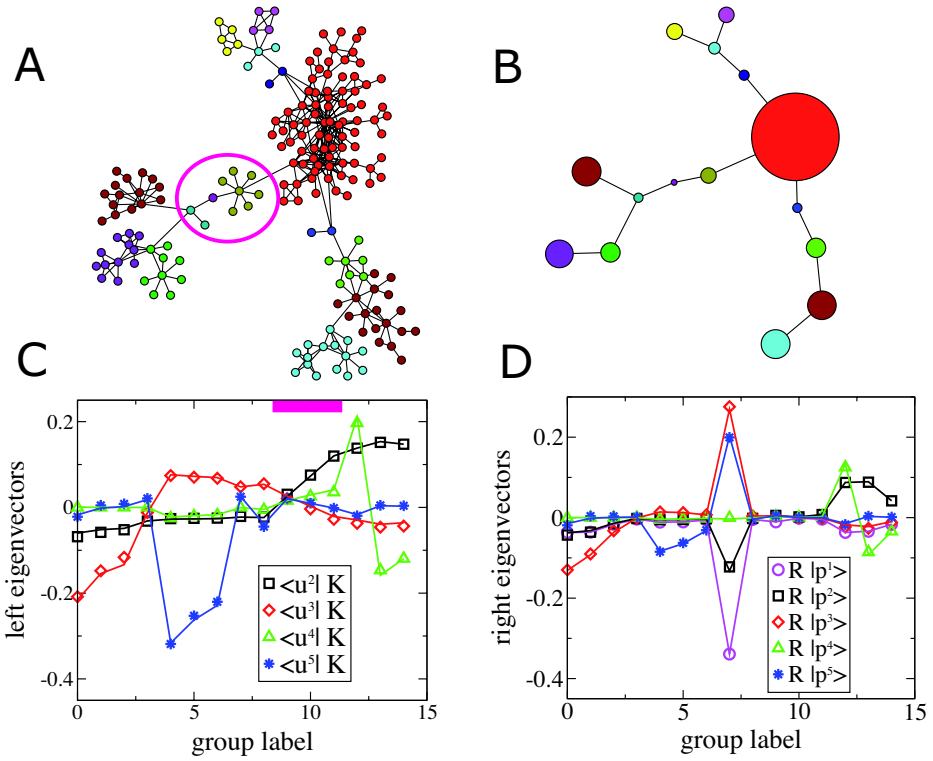


Figure 7.19: **A**: Original network, colors represent the different groups obtained after relaxing the coarse graining. **B**: reduced network. The first 4 non-trivial left eigenvectors have been used in the coarse graining with $I = 20$. **C**: first 4 non-trivial left eigenvectors. **D**: first 5 right eigenvectors, the first one being the stationary state. Symbols in **C** (resp. **D**) show the projection $\langle u^\alpha | K$ (resp. $R|p^\alpha\rangle$). Continuous lines represent the eigenvectors $\langle \tilde{u}^\alpha |$ (resp. $|\tilde{p}^\alpha\rangle$) of $\tilde{W}$. For clarity $\langle u^1 | K$ and $\langle \tilde{u}^1 |$ are not shown since both are flat.

7.8 Extension to symmetric matrices

Random walks on complex networks represent a specific class of dynamical processes. Sometimes they account for real processes taking place on networks, such as web crawls, traffic flow or peptide dynamics in configuration space networks. However, not all complex networks underlie such kind of dynamics. On the one hand random walks on cell cycle or synonymy networks do not correspond to any existing physical process and should merely be considered as a useful tool to study these networks. On the other hand several other kinds of dynamics take place on networks for which stochastic matrices are not relevant, such as percolation [72], synchronization of coupled oscillators [132], Boolean networks, Gaussian networks [52, 9], etc. Therefore a useful issue is to extend the framework of Spectral Coarse Graining so that it is no longer restricted to stochastic matrices and random walks.

In this Chapter, we show that Spectral Coarse Graining can be extended to symmetric matrices associated with a network. This is a crucial step forward enlarging the field of applications of coarse graining ideas to a much wider range of dynamical processes on networks. In this extension, the equivalence between preserving eigenvalues and grouping nodes summing up their edges does not hold exactly, but requires to update in a suitable way the edge weight. However this update turns out to be very intuitive when considering networks underlying dynamical processes described by the Laplacian matrix (see section 7.8.3).

For the sake of clarity, the mathematical framework is first presented in the case of adjacency matrices A. In a symmetric network, eigenvalues of A are real and can take positive or negative values. The largest eigenvalue is related to a variety of dynamical processes on networks, such as percolation on directed networks [72] or critical coupling strength for a collection of chaotic systems [152]. Recently the sensitivity of the largest eigenvalue upon node removal has been used to characterize the dynamical importance of nodes [153]. For these reasons the largest eigenvalues of A are the most relevant ones and will be considered in the SCG strategy. Later on, in Section 7.8.3, the Laplacian L is used to coarse grain Gaussian Networks. In this case the lowest eigenvalues of L bear the relevant properties of the network dynamics. Finally we design by $|v^\alpha\rangle$ the eigenvectors of symmetric matrices (the distinction between left and right eigenvectors is no longer necessary), while $\langle u^\alpha|$ and $|p^\alpha\rangle$ were left and right eigenvectors of stochastic matrices.

7.8.1 Mathematical framework

The starting point of Spectral Coarse Graining based on the stochastic matrix W was the observation that two nodes having exactly the same neighbors have the same components along the left eigenvectors of W with non-zero eigenvalues. Considering the adjacency matrix A of a symmetric network, the property still holds. If two nodes (1 and 2) have exactly the same neighbors, the eigenvector components v_1^α and v_2^α are the same for all α with $\lambda^\alpha \neq 0$. Moreover a zero eigenvalue is associated with each pair of nodes having the same neighbors since lines 1 and 2 of A are equal. As before, we can define ideal groups as nodes having the same components along an eigenvector $|v^\alpha\rangle$ (the distinction between left and right eigenvectors is no longer necessary since the matrix is symmetric).

The crucial question is to know how nodes should be grouped in order to preserve the spectral properties and the symmetry of the adjacency matrix. For simplicity, we still consider the case where $v_1^\alpha = v_2^\alpha$, implying that nodes 1 and 2 are grouped together. Intuitively, we could merge them summing up all their edges, as in the case of SCG based on the stochastic matrix. In terms of the projection operators R and K, it implies that R remains the same as in Eq. (7.3), whereas $K = R^T$. However, a simple calculation shows that $R^T R|v^\alpha\rangle = (v_1^\alpha + v_2^\alpha, v_1^\alpha + v_2^\alpha, v_3^\alpha, \ldots)^T \neq |v^\alpha\rangle$, even if $v_1^\alpha = v_2^\alpha$. Thus if nodes are simply merged, $R|v^\alpha\rangle$ (resp. λ^α) is not an eigenvector (resp. an eigenvalue) of $\tilde{A} = RAR^T$ and the goal of the coarse graining is

α	λ^α	$\tilde{\lambda}^\alpha$	$\frac{\langle\tilde{v}^\alpha\|P\|v^\alpha\rangle}{\|\|P\|v^\alpha\rangle\|\|\cdot\|\|\tilde{v}^\alpha\|\|}$
1	$5.3111{\cdot}10^5$	$5.3107{\cdot}10^5$	1
2	$3.872{\cdot}10^5$	$3.869{\cdot}10^5$	0.99998
3	$3.274{\cdot}10^5$	$3.271{\cdot}10^5$	0.99996

Table 7.5: Columns 1 and 2: the three largest eigenvalues of A for the di-alanine network. Column 3: Scalar product between $P|v^\alpha\rangle$ and $|\tilde{v}^\alpha\rangle$ for the right eigenvectors.

not reached. Furthermore, using R and K as defined in Eq. (7.3) does not yield a symmetric $\tilde{A}$, which leads to a pathological situation since a symmetric network should transform into another symmetric network. The two conditions required above can be satisfied defining the new matrix P as:

$$P = \begin{pmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & 0\dots 0 \\ 0 & 0 & \\ \vdots & \vdots & I_{N-2} \\ 0 & 0 & \end{pmatrix} \quad \text{and} \quad P^T = \begin{pmatrix} \frac{1}{\sqrt{2}} & 0 & \dots & 0 \\ \frac{1}{\sqrt{2}} & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & I_{N-2} & \\ 0 & & & \end{pmatrix} \tag{7.11}$$

With this definition, the matrix $\tilde{A} = PAP^T$ is symmetric and can be considered as the adjacency matrix of a reduced network.As required $PP^T = I_{\tilde{N}}$ and $P|v^\alpha\rangle$ is an eigenvector of $\tilde{A}$ with eigenvalue $\tilde{\lambda}^\alpha = \lambda^\alpha$.

If groups consist in more than 2 nodes P is given by

$$P_{Ci} = \frac{1}{\sqrt{|C|}}\delta_{C,i}$$

with $C = 1, \dots, L$ the label of the groups and $|C|$ the number of nodes in group C. $\delta_{C,i}$ is defined as one if i belongs to group C, 0 otherwise.

Replacing R, resp. K, by P, resp. P^T, the same perturbation approach as in Section 7.3 can be carried out, as recently shown by Morton [114]. If the components of $|v^\alpha\rangle$ are very close within each group (i.e. $|v^\alpha\rangle = P^T P|v^\alpha\rangle + |\epsilon\rangle$), $AP|v^\alpha\rangle$ differs from $\lambda^\alpha P|v^\alpha\rangle$ by a vector whose norm scales as $|||\epsilon\rangle||$. From a more general point of view, non-zero elements in P can be defined in other ways as long as $PP^T = I_{\tilde{N}}$. In this case the condition for the exact coarse graining becomes $P^T P|v^\alpha\rangle = |v^\alpha\rangle$ and groups are no longer defined as nodes whose eigenvector components are equal. An example of this case is presented in the next Section.

coarse graining networks along the eigenvectors of A performs well and the perturbation approach is robust. As an example, we performed the coarse graining along the three first eigenvectors corresponding to the largest eigenvalues of the di-alanine network adjacency matrix. Results are displayed in Table 7.5.

Spectral properties of A are often strongly influenced by the distribution of weights in the network, resulting in very large eigenvalues as in Table 7.5. A common way of circumventing this problem is to consider either a symmetric matrix similar to the stochastic matrix W or the Laplacian matrix L (see Section 3.2.7) instead of A. We will describe our main results for these two matrices in the next two Sections.

7.8.2 Connection with stochastic matrices

The stochastic matrix W is usually defined from the adjacency matrix as:

$$W = AD^{-1},$$

with $D_{ii} = \sum_l A_{li}$ a diagonal matrix. Interestingly, W can also be expressed as

$$W = D^{1/2} M D^{-1/2}, \tag{7.12}$$

with M a symmetric matrix if A is symmetric. Since M and W are similar, they have exactly the same eigenvalues $1 = \lambda^1 \geq \lambda^2 \geq \ldots \geq \lambda^N \geq -1$. Eigenvectors $|v^\alpha\rangle$ of M are then given by $|v^\alpha\rangle = D^{-1/2}|p^\alpha\rangle = \left(\langle u^\alpha | D^{1/2}\right)^T$, with $|p^\alpha\rangle$ and $\langle u^\alpha|$ the right and left eigenvectors of W.

Applying R and K on both sides of Eq. (7.12) gives

$$W = D^{1/2} M D^{-1/2} \Leftrightarrow \tilde{W} = R D^{1/2} M D^{-1/2} K \tag{7.13}$$
$$\tag{7.14}$$

In the coarse-grained network, we can further define the diagonal matrix $\tilde{D}$ as $\tilde{D}_{CC} = \sum_{i \in C} D_{ii}$. With this definition, the coarse-grained matrix $\tilde{M}$ is expressed as

$$\tilde{M} = \tilde{D}^{-1/2} \tilde{W} \tilde{D}^{1/2} = \tilde{D}^{-1/2} R D^{1/2} M D^{-1/2} K \tilde{D}^{1/2} \tag{7.15}$$

A simple calculation shows that if nodes 1 and 2 are merged, matrix $\tilde{D}^{-1/2} R D^{1/2}$ reads

$$\tilde{D}^{-1/2} R D^{1/2} = \begin{pmatrix} \sqrt{\frac{k_1}{k_1+k_2}} & \sqrt{\frac{k_2}{k_1+k_2}} & 0 \ldots 0 \\ 0 & 0 & \\ \vdots & \vdots & I_{N-2} \\ 0 & 0 & \end{pmatrix} = \left(D^{-1/2} K \tilde{D}^{1/2}\right)^T = P,$$

where the third equality holds since $p_i^1 \propto k_i$ for undirected networks. More generally we have that:

$$P_{Ci} = (\tilde{D}^{-1/2} R D^{1/2})_{Ci} = \delta_{Ci} \sqrt{\frac{k_i}{\sum_{j \in C} k_j}},$$

with δ_{Ci} defined as 1 if node i belongs to group C and 0 otherwise. As expected $PP^T = I_{\tilde{N}}$, which ensures that P is properly defined. Therefore SCG of W with R and K defined in Eq. (7.3) is equivalent to SCG of the symmetric matrix M with P and P^T. In this case the exact SCG groups nodes such that $P^t P |v^\alpha\rangle = |v^\alpha\rangle$, that is $\frac{v_i^\alpha}{k_i} = \frac{v_j^\alpha}{k_j} \Leftrightarrow u_i^\alpha = u_j^\alpha$ as expected.

Another interesting feature of the similarity relation between W and M concerns the eigenvalues of $\tilde{M}$ or equivalently of $\tilde{W}$. In all examples studied previously (Tables 7.2, 7.3, 7.4, 7.5), we have seen that $\lambda^\alpha \geq \tilde{\lambda}^\alpha$. Here we give a mathematical proof for this result by considering the eigenvalues of M and $\tilde{M}$. Since M is symmetric, the Courant-Fischer theorem [80] allows to express the eigenvalues of M as:

$$\lambda^\alpha = \min_{s^1, \ldots, s^{\alpha-1} \in \mathbb{R}^N} \quad \max_{\substack{x \perp s^1, \ldots, s^{\alpha-1} \\ x^T x = 1, x \in \mathbb{R}^N}} x^T M x, \tag{7.16}$$

with $s^1, \ldots, s^{\alpha-1}$ any set of $\alpha - 1$ vectors. Although Eq. (7.19) is of little use for computing the eigenvalues it will be very useful to find an upper bound for $\tilde{\lambda}^\alpha$. We have the following inequality:

$$\begin{aligned}
\max_{\substack{x \perp s^1,\ldots,s^{\alpha-1} \\ x^T x = 1, x \in \mathbb{R}^N}} x^T M x &\geq \max_{\substack{x \perp s^1,\ldots,s^{\alpha-1} \\ x^T x = 1, x = P^T y, y \in \mathbb{R}^{\tilde{N}}}} x^T M x \\
&= \max_{\substack{y \perp Ps^1,\ldots,Ps^{\alpha-1} \\ y^T y = 1}} y^T P M P^T y \\
&= \max_{\substack{y \perp Ps^1,\ldots,Ps^{\alpha-1} \\ y^T y = 1}} y^T \tilde{M} y, \qquad (7.17)
\end{aligned}$$

The inequality stems from the restriction to a smaller subspace for the choice of x. In the second line we have used that $PP^T = I_{\tilde{N}}$. Since Eq. (7.17) holds for any choice of $s^1, \ldots, s^{\alpha-1}$ as long as $\alpha \leq \tilde{N}$, we can take the minimum on both sides:

$$\begin{aligned}
\min_{s^1,\ldots,s^{\alpha-1} \in \mathbb{R}^N} \max_{\substack{x \perp s^1,\ldots,s^{\alpha-1} \\ x^T x = 1}} x^T M x &\geq \min_{s^1,\ldots,s^{\alpha-1} \in \mathbb{R}^N} \max_{\substack{y \perp Ps^1,\ldots,Ps^{\alpha-1} \\ y^T y = 1}} y^T \tilde{M} y \\
\Leftrightarrow \lambda^\alpha &\geq \tilde{\lambda}^\alpha. \qquad (7.18)
\end{aligned}$$

where we have used that for any set $\{\tilde{s}^1, \ldots, \tilde{s}^{\alpha-1}\}$ of $\alpha - 1$ vectors in $\mathbb{R}^S$, there exists a set of vectors $\{s^1, \ldots, s^{\alpha-1}\}$ in $\mathbb{R}^N$ such that $\tilde{s}^\beta = Ps^\beta$. The proof given above is valid for any symmetric matrix and for any $\tilde{N} \times N$ matrix P satisfying $PP^T = I_{\tilde{N}}$. In particular Eq. (7.18) holds for any choice of the groups. Now, since M and W, as well as $\tilde{M}$ and $\tilde{W}$ have exactly the same eigenvalues, Eq. (7.18) shows that considering SCG of stochastic matrices, eigenvalues of $\tilde{W}$ are always smaller than the one of W. It will be interesting to investigate whether this result can be extended to directed networks for which the Courant-Fischer theorem does not hold.

Finally we note that the Courant-Fischer theorem [80] can also be expressed in a more appropriate form when considering low eigenvalues (large α):

$$\lambda^\alpha = \max_{s^1,\ldots,s^{N-\alpha} \in \mathbb{R}^N} \min_{\substack{x \perp s^1,\ldots,s^{N-\alpha} \\ x^T x = 1, x \in \mathbb{R}^N}} x^T M x, \qquad (7.19)$$

Applying the same argument as in Eq. (7.17) and (7.18), we have for $\alpha > N - \tilde{N}$ that:

$$\lambda^\alpha \leq \tilde{\lambda}^{\alpha - N + \tilde{N}} \qquad (7.20)$$

We stress that Eq. (7.18) and Eq. (7.20) applies to a different range of αs and cannot be combined. Eq. (7.20) shows that comparing the eigenvalues of the coarse-grained network with the lower eigenvalues of the initial network yields larger eigenvalues for the coarse-grained network. This result will be especially interesting for the Laplacian matrix L since the lowest eigenvalues are the most relevant ones in many dynamical processes described by L (see Table 7.6).

7.8.3 Laplacian matrix and Gaussian Network Model (GNM)

The Laplacian matrix is defined as $L = D - A$, where A is the adjacency matrix (taken as symmetric) and D is the diagonal matrix with $D_{ii} = \sum_j A_{ij}$. In simple networks, $D_{ii} = k_i$. All lines of L sum up to 0, which implies that $|v^N\rangle = (1, \ldots, 1)$ is an eigenvector with eigenvalue 0. Moreover all other eigenvalues are larger than 0 and the lowest ones play a similar role as the largest eigenvalues of W. Thus, in the perspective of SCG, the lowest eigenvalues of L are

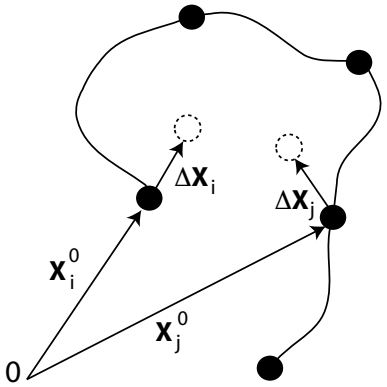


Figure 7.20: Schematic representation of the fluctuations described in the GNM model.

the ones that should be preserved. Since L is a symmetric matrix, the same coarse graining formalism as in Section 7.8.1 can be applied (see also Appendix B.2 for a discussion about how redundant information is removed from the network in the coarse graining based on the Laplacian).

The interest in the Laplacian stems from several useful properties of L. Already more than hundred years ago, Kirchhoff noticed that L describes the voltage of each node in a resistor network. More recently, the lowest non-trivial eigenvalues of L were shown to play a critical role in synchronization of complex networks [132], community structure detection [51, 184, 38], fluctuations in Gaussian networks [77], and several other dynamical processes on networks. To avoid a plethora of different examples, we focus on SPC of Gaussian networks in this Chapter.

The Gaussian Network Model dates back to the seminal work of Flory [52] about polymers. Polymers are long chains of monomers that assemble in a linear way. In order to describe the dynamics of such systems, Flory introduced the approximation that distances separating two monomers are distributed in a Gaussian way. Approximatively 20 years later the ideas of Flory were extended to the description of proteins by Bahar *et al.* who introduced the Gaussian Network Model (GNM) [9, 77]. In this model, only C_α atoms are taken into account and they act as indistinguishable beads of mass m. However, contrary to the work of Flory, not only interactions along the protein backbone are considered, but the native structure of the protein is incorporated in the model. The most intuitive way of including information about the native structure of a protein is to assume that C_α atoms "feel each other" if their respective distance is smaller than a certain cut-off θ in the native state. The resulting interaction network topology is therefore no longer a chain, but reflects the properties of the contact map.

The Gaussian distribution of the fluctuations from equilibrium position have been represented by the following potential [8],

$$V = \frac{\gamma}{2} \sum_{ij} A_{ij} \left((\mathbf{X}_i - \mathbf{X}_i^0) - (\mathbf{X}_j - \mathbf{X}_j^0) \right)^2 , \tag{7.21}$$

where $\mathbf{X}_i$ is the vector position of the i^{th} C_α atom and $\mathbf{X}_i^0$ its position at equilibrium (see Figure 7.20). A_{ij} stands for the presence or not of an interaction ($A_{ij} = 1$ if $|\mathbf{X}_i^0 - \mathbf{X}_j^0| < \theta$, 0 otherwise). As expected the system is invariant under a global translation. Two main observations arise from Eq. (7.21). First all interactions have the same coupling given by γ. Second, the fluctuations $\Delta\mathbf{X}_i = (\mathbf{X}_i - \mathbf{X}_i^0)$ follow an isotropic Gaussian distribution. This means that $\Delta\mathbf{X}_i - \Delta\mathbf{X}_j$ is equally distributed in all directions, no matter whether it is parallel or perpendicular to the vector $\mathbf{X}_i^0 - \mathbf{X}_j^0$. In this sense GNM assumes a completely isotropic

motion of the C_α. In particular it should be stressed that GNM is *not* equivalent to a network of springs with force constant γ. More recently, GNM have been extended into the Anisotropic Network Model (ANM) in order to take into account the anisotropy of motion [8].

The global isotropy of GNM yields important consequences. In particular there is no need to distinguish between some particular direction of motion for C_α atoms. For this reason fluctuations $\Delta\mathbf{X}_i$ are often considered as a scalar ΔX_i representing the global fluctuation from equilibrium position. Within this framework, the dynamics of GNM is described by the following Hamiltonian:

$$\mathcal{H}(X) = \frac{m}{2}\sum_i \dot{X_i}^2 + \frac{\gamma}{2}\sum_{ij} A_{ij}\left(\Delta X_i - \Delta X_j\right)^2$$

which can be rewritten as,

$$\mathcal{H}(\Delta X) = \frac{1}{2}\dot{\Delta X}^T M \dot{\Delta X} + \frac{\gamma}{2}\Delta X^T L \Delta X$$

since $\dot{X} = \dot{\Delta X}$. $M = mI_{N\times N}$ and L stands for the Laplacian of the network defined by the contact map of the protein with a cut-off θ. Using normal coordinates, $\Delta S = M^{1/2}\Delta X$, we eventually obtain:

$$\mathcal{H}(\Delta S) = \frac{1}{2}\dot{\Delta S}^T \dot{\Delta S} + \frac{\gamma}{2}\Delta S^T M^{-1/2} L M^{-1/2}\Delta S \tag{7.22}$$

Normal Mode Analysis yields $N-1$ different oscillatory modes with intrinsic frequencies given by $\frac{\gamma}{m}\lambda^\alpha$ [3], λ^α standing for the eigenvalues of L. In particular the slow modes are given by the lowest eigenvalues of L.

GNM dynamics is particularly suited for a coarse graining approach since the entire information about interactions is contained in the network topology, or equivalently in the matrix L. As usual, we first assume the ideal case in which nodes 1 and 2 have exactly the same neighbors in the network of interactions between C_α atoms. If $|v^\alpha\rangle$ is an eigenvector of L, then $v_1^\alpha = v_2^\alpha$ for non-pathological eigenvalues (see Appendix B.2). As for SCG based on the stochastic matrix, the natural strategy is to merge these two nodes, keeping all their interactions. This operation is carried out by the matrix R of Eq. (7.3), defined as

$$R = \begin{pmatrix} 1 & 1 & 0\ldots 0 \\ 0 & 0 & \\ \vdots & \vdots & I_{N-2} \\ 0 & 0 & \end{pmatrix}.$$

The Laplacian of the reduced network is given by $\tilde{L} = RLR^T$. Similarly $\tilde{M} = RMR^T$, since masses have to sum up when merging C_α atoms. Therefore the dynamics of GNM on the reduced network is described by the following Hamiltonian:

$$\tilde{\mathcal{H}}(\Delta\tilde{S}) = \frac{1}{2}\dot{\Delta\tilde{S}}^T \dot{\Delta\tilde{S}} + \frac{\gamma}{2}\Delta\tilde{S}^T (RMR)^{-1/2} RLR^T (RMR)^{-1/2}\Delta\tilde{S} \tag{7.23}$$

We stress here that since GNM is fully described by the network topology, merging nodes can be done without taking into account the spatial position of the corresponding C_α. The only requirement is to preserve all interactions identified in the initial system and to sum the masses.

Two interesting features emerge now. First if $|v^\alpha\rangle$ is an eigenvector of L with eigenvalue λ^α, then $|v^\alpha\rangle$ is an eigenvector of $M^{-1/2}LM^{-1/2}$ with eigenvalue $\mu^\alpha = \frac{\lambda^\alpha}{m}$. Second, the matrix describing the potential energy in Eq. (7.23) can be rewritten as:

$$\begin{aligned}(RMR)^{-1/2}RLR^T(RMR)^{-1/2} &= (RMR)^{-1/2}RM^{1/2}M^{-1/2}LM^{-1/2}M^{1/2}R^T(RMR)^{-1/2} \\ &= PM^{-1/2}LM^{-1/2}P^T,\end{aligned}$$

[3] Note that usually m is omitted since all masses are assumed to be equal.

with P defined as in Eq. (7.11). Therefore merging nodes with $v_i^\alpha = v_j^\alpha$ is equivalent to coarse graining the matrix $M^{-1/2}LM^{-1/2}$ into $PM^{-1/2}LM^{-1/2}P^T$ as it was defined in Section 7.8.1 for symmetric matrices. In particular it ensures that the eigenvalue $\mu^\alpha = \frac{\lambda^\alpha}{m}$ is preserved in the reduced network.

To summarize, we have shown that the coarse-grained network in which nodes (i.e. C_α atoms) with similar eigenvector components have been merged has the same spectral properties as the initial one, considering the dynamics induced in GNM. In particular our derivation ensures that the slow modes are preserved in the reduced network.

Despite its simplicity and its isotropy assumption, one of the successes of GNM is the possibility to compute analytically the correlations $< \Delta X_i \cdot \Delta X_j >$. These correlations were shown to reproduce with a reasonable accuracy the temperature factors given in the Protein Database [77], which are thought to account for fluctuations from equilibrium position. In the framework of GNM, correlations are computed as:

$$\begin{aligned} < \Delta X_i \cdot \Delta X_j > &= m^{-1} < \Delta S_i \cdot \Delta S_j > \\ &= m^{-1} \frac{1}{Z} \int \Delta S_i \cdot \Delta S_j e^{-\mathcal{H}(\Delta S)/k_B T} \mathrm{d}\{\Delta S\} \mathrm{d}\{\dot{\Delta S}\} \end{aligned} \tag{7.24}$$

with $Z = \int e^{-H(\Delta s)/kT} \mathrm{d}\{\Delta S\} \mathrm{d}\{\dot{\Delta S}\}$. We note that in Eq. (7.24) the integral is performed over the $N-1$ dimensional subspace perpendicular to the eigenvector $|p^N\rangle$ associated with $\lambda^N = 0$ and corresponding to global translations. Since the Hamiltonian does not mix terms involving ΔS and $\dot{\Delta S}$, the contribution of the kinetic energy vanishes with the denominator. Using the spectral decomposition of $M^{-1/2}LM^{-1/2}$, one shows that,

$$< \Delta S_i \cdot \Delta S_j >= \frac{k_B T}{\gamma} \sum_{\alpha=1}^{N-1} \frac{1}{\mu^\alpha} v_i^\alpha v_j^\alpha \tag{7.25}$$

In the coarse-grained system described by the Hamiltonian of Eq. (7.23), correlations are given by:

$$< \Delta \tilde{S}_A \cdot \Delta \tilde{S}_B >= \frac{k_B T}{\gamma} \sum_{\alpha=1}^{L-1} \frac{1}{\tilde{\mu}^\alpha} \tilde{v}_A^\alpha \tilde{v}_B^\alpha = \frac{k_B T}{\gamma} \sum_{\alpha=1}^{N-1} \sum_{i \in A, j \in B} \frac{1}{\mu^\alpha} v_i^\alpha v_j^\alpha \frac{1}{\sqrt{|A||B|}} \tag{7.26}$$

The second equality was obtained replacing $\tilde{v_A^\alpha}$ by $(P|v^\alpha\rangle)_A = \frac{1}{\sqrt{|A|}} \sum_{i \in A} v_i^\alpha$ and μ^α by $\tilde{\mu}^\alpha$, which is valid if the coarse graining is exact. The sum can be done over the $N-1$ eigenvectors since eigenvalues that had been removed in the exact coarse graining do not contribute to the sum (see Appendix B.2). From the previous expression, we can compute the quantities $< \Delta \tilde{X}_A \cdot \Delta \tilde{X}_B >$ for the coarse-grained network:

$$
\begin{aligned}
< \Delta\tilde{X}_A \cdot \Delta\tilde{X}_B > &= < \Delta\tilde{S}_A \cdot \Delta\tilde{S}_B > \frac{1}{\sqrt{m_A m_B}} \\
&= \frac{1}{m}\frac{1}{|A||B|}\frac{k_B T}{\gamma}\sum_{\alpha=1}^{N-1}\sum_{i\in A, j\in B}\frac{1}{\mu^\alpha}v_i^\alpha v_j^\alpha \\
&= \frac{1}{|A||B|}\frac{k_B T}{\gamma}\sum_{\alpha=1}^{N-1}\sum_{i\in A, j\in B}\frac{1}{\lambda^\alpha}v_i^\alpha v_j^\alpha \\
&= \frac{1}{|A||B|}\sum_{i\in A, j\in B} < \Delta X_i \cdot \Delta X_j > \\
&= < \left(\frac{1}{|A|}\sum_{i\in A}\Delta X_i\right)\cdot\left(\frac{1}{|B|}\sum_{j\in B}\Delta X_j\right) > \qquad (7.27)
\end{aligned}
$$

Eq. (7.27) shows that the coarse-grained network in which nodes have merged is exactly equivalent to the initial one considering fluctuations $\Delta\tilde{X}_A = \frac{1}{|A|}\sum_{i\in A}\Delta X_i$, which correspond to the expected fluctuations when observing the network at a larger scale. In particular the update in the edge weight to coarse grain symmetric matrices turns out to naturally account for the effect of summing the mass of each bead, as long as all beads have the same initial mass m. In the case of almost exact coarse graining preserving only the lowest eigenvalues, the previous derivation shows that the slow modes of the dynamics induced by GNM are preserved in the reduced network.

7.8.4 Immunoglobuline Gaussian Network

We applied SCG on the Gaussian Network extracted form the immunoglobuline structure. The protein is made of 1316 amino acids and consists of three clear domains, as shown in the onset of Figure 7.21A. A threshold $\theta = 8\mathring{A}$ has been chosen to define interactions between C_α atoms.

The spectrum of L is characterized by two very small eigenvalues ($\lambda^{N-1} = 2.21 \cdot 10^{-3}$ and $\lambda^{N-2} = 2.76 \cdot 10^{-3}$), accounting for the two slowest modes of the Gaussian network (here we have fixed $m = 1$ and $\frac{k_B T}{\gamma} = 1$ for simplicity). We used the two corresponding eigenvectors to coarse grain the network. Taking $I = 100$, we obtained a reduced network of $\tilde{N} = 83$ nodes with $\lambda^{N-1} = 2.28 \cdot 10^{-3}$ and $\lambda^{N-2} = 2.85 \cdot 10^{-3}$. Thus the slow modes are well preserved in the reduced network

More interesting results could be obtained by including other eigenvectors in the coarse graining. In order to give a larger importance to the slowest modes, we used different numbers of intervals along the eigenvectors as already discussed in Section 7.7. Along $|v^{N-1}\rangle$ and $|v^{N-2}\rangle$, $I = 100$ intervals have been defined. Then for each new eigenvector included in the coarse graining, we divided by 2 the number of intervals (see Table 7.6). For instance $I = 25$ intervals have been taken along $|v^{N-4}\rangle$. As predicted by Eq. 7.20 eigenvalues of the coarse-grained network are larger than those of the initial network. In Figure 7.21, we display the initial and the coarse-grained network of immunoglobuline ($N = 1316$ and $\tilde{N} = 251$). Colors reflect the three domains of immunoglobuline. The onset of Figure 7.21A represents a spatial visualization of the protein, considering each C_α as a bead (for clarity interactions have been removed). The nine lowest non-trivial eigenvectors have been considered in the coarse graining. In Figure 7.21B, node size is proportional to the number of nodes in each group. In the coarse-grained protein shown in the onset, the position of each bead is given by the center of mass of C_α atoms within each group. A visual inspection of the coarse-grained protein already indicates that results are reasonable. Table 7.6 provides a more quantitative confirmation by showing that the coarse

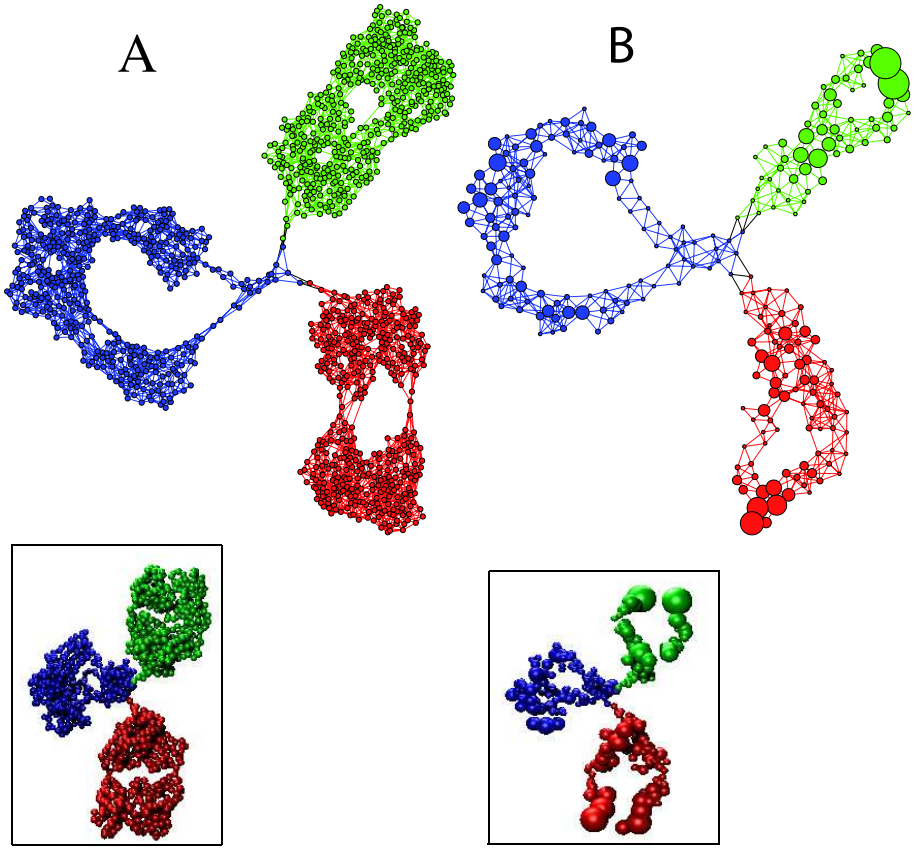


Figure 7.21: **A**: Network of interactions between C_α in the immunoglobuline. Onset: Spatial representation of the protein, each bead representing a C_α atom as in the native state. **B**: Coarse-grained network. Node size is proportional to the group size. Onset: Coarse-grained protein. The position of each bead is given by the center of mass of the C_α atoms of each group.

graining leaves unchanged the slowest modes of the dynamics, as long as a reasonable number of intervals are used to partition the eigenvectors.

From the point of view of correlations, the exact coarse graining was shown in Eq. (7.27) to leave unchanged $< \Delta\tilde{X}_A \cdot \Delta\tilde{X}_B >$. In the perturbation approach the exact equality does not hold any more, since groups no longer consist of nodes with the same fluctuations. In particular we expect to see slightly lower auto-correlations $< \Delta\tilde{X}_A \cdot \Delta\tilde{X}_A >$ in the reduced network since internal interactions are smoothed out in the coarse graining. In Figure 7.22 the comparison between $< \Delta\tilde{X}_A \cdot \Delta\tilde{X}_A >$ and $< \left(\frac{1}{|A|}\sum_{i\in A}\Delta X_i\right) \cdot \left(\frac{1}{|A|}\sum_{j\in A}\Delta X_j\right) >$ is displayed for each group. As expected, the correlations are slightly lower in the reduced network. However the global behavior is very well recovered. Finally Figure 7.23 shows the correlations $< \Delta\tilde{X}_A \cdot \Delta\tilde{X}_B >$ and $< \left(\frac{1}{|A|}\sum_{i\in A}\Delta X_i\right) \cdot \left(\frac{1}{|B|}\sum_{j\in B}\Delta X_j\right) >$ as a function of B for three different groups A randomly chosen in the three domains.

In general, Figure 7.22 and 7.23 show that the global behavior of correlations remains unchanged under coarse graining, as well as the slow modes (Table 7.6). These results indicate that the dynamics described by GNM can naturally be coarse-grained such that most features of interest are preserved under spectral coarse graining. Since correlations computed in the coarse-grained network approximate very well the initial ones, the coarse grained network can even be used to make predictions about the initial one. From a computational point of view, this result is extremely interesting since the coarse graining only requires to compute a few

α	I	λ^α	$\tilde{\lambda}^\alpha$
N-1	100	0.00221	0.00223
N-2	100	0.00276	0.00279
N-3	50	0.0360	0.0369
N-4	25	0.0382	0.0409
N-5	12	0.0711	0.0776
N-6	6	0.0785	0.087
N-7	3	0.2029	0.208
N-8	2	0.2154	0.342
N-9	2	0.2213	0.418

Table 7.6: Column 1: Eigenvalue label. Column 2: Number of intervals defined along the eigenvectors $|v^\alpha\rangle$. Column 3: Eigenvalues λ^α in the initial network. Column 4: Eigenvalues $\tilde{\lambda}^\alpha$ in the reduced network.

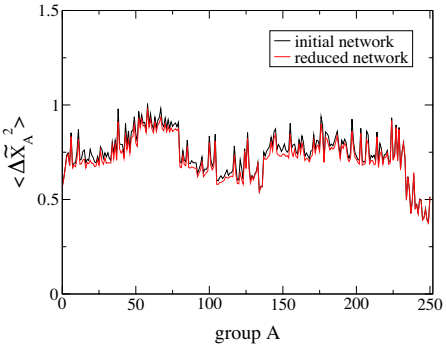


Figure 7.22: Comparison between $< \Delta\tilde{X}_A \cdot \Delta\tilde{X}_A >$ and $< \left(\frac{1}{|A|}\sum_{i\in A}\Delta X_i\right)\cdot\left(\frac{1}{|A|}\sum_{i\in A}\Delta X_i\right) >$ for each group of the coarse-grained network of Figure 7.21.

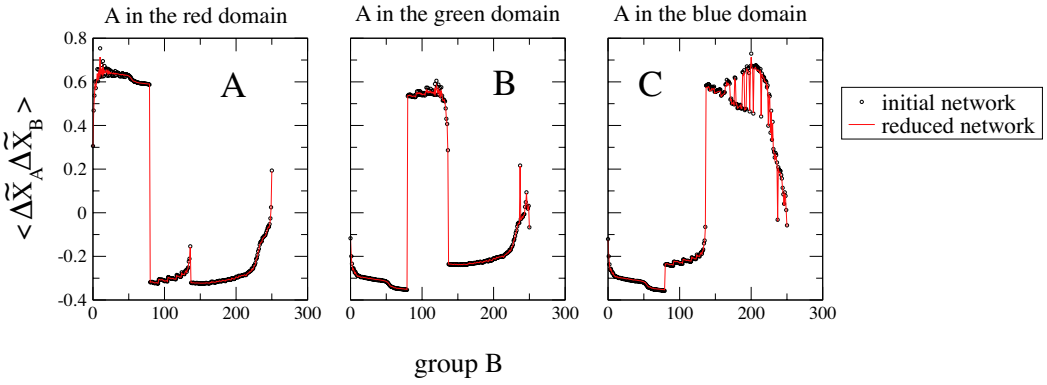


Figure 7.23: Comparison between $< \Delta\tilde{X}_A \cdot \Delta\tilde{X}_B >$ and $< \left(\frac{1}{|A|}\sum_{i\in A}\Delta X_i\right)\cdot\left(\frac{1}{|B|}\sum_{j\in B}\Delta X_j\right) >$ for each group B. **A**: Group A was chosen in the red domain. **B**: group A was chosen in the green domain. **C**: group A was chosen in the blue domain.

eigenvectors, which can be done easily even for large proteins.

Another remarkable feature emerging from Figure 7.21 concerns the group size. While large groups are found within the three domains, nodes lying in the middle of the network, that is in the region where the three domains are attached together, are extremely well preserved and most nodes of the coarse-grained network are made of one single node of the original one. Indeed these nodes are crucial for the protein dynamics and act as fulcrum between the three domains. We therefore suggest that the analysis of the nodes most preserved under SCG may be a way of identifying nodes with the highest dynamical importance in the protein.

7.8.5 Related works

The question of preserving the slow vibrational modes of proteins in GNM has already been discussed in [94] considering a structural coarse graining approach in which C_α close to each other are considered as one single unit. Compared to structural coarse graining, SCG has the advantage of ensuring that slow vibrational modes are automatically preserved. Indeed results of SCG are consistent with the protein structure in the sense that C_α lying very far from each other are not grouped together, as shown in Figure 7.21. In addition SCG allows to distinguish between regions in which the protein can be easily coarse grained, and regions in which no coarse graining can be applied without altering the slow modes.

Another related method has been designed in [31]. In this work the authors introduce a coarse graining based on a stochastic process on the network of interactions, yet quite different from the Spectral Coarse Graining discussed in Chapter 7. Unfortunately their method does not ensure to preserve the slow modes of the dynamics which, in our opinion, is a crucial feature. Moreover we believe that stochastic processes on Gaussian Networks are not very natural, whereas SCG considers the Laplacian matrix, which is the central feature of GNM.

As a final remark, we stress that GNM is an extremely simplified model. Nevertheless, a global coarse graining approach aiming at reducing the complexity of real proteins is likely to leave unchanged the slow modes described in GNM. SCG presented here satisfies automatically this condition and offers a new candidate to coarse grain proteins based on their dynamics rather than on their structure. In particular it allows immediately to identify which amino acids are crucial from a dynamical point of view, and which can be grouped without altering the dynamics. In order to extend this result, it will be interesting to check whether the coarse-grained protein performs similarly under more realistic dynamics, such as ANM [8], Go model [67], or even Molecular Dynamics simulations. Another particular promising extension would be to start from the fluctuations observed in molecular dynamics simulations, to build a quadratic Hamiltonian accounting for these fluctuation (which can be done by simply inverting the correlation matrix as in [145]) and to apply the coarse graining on this model.

7.9 Perspectives and conclusions

In the second part of this Thesis, we have investigated a new strategy to reduce the complexity of networks. This strategy focuses on the information contained in the first eigenvectors of either the stochastic matrix or the Laplacian, akin to Principle Component Analysis. Yet it differs from Principle Component Analysis since we do not simply project the matrix on the S first eigenvectors, which would be equivalent to truncate the spectral decomposition to the S first terms ($W = \sum_{\alpha=1}^{N} \lambda^\alpha |p^\alpha\rangle\langle u^\alpha| \rightarrow \sum_{\alpha=1}^{S} \lambda^\alpha |p^\alpha\rangle\langle u^\alpha|$). Instead of that, we consider the S-dimensional embedding space in which each node is represented by its corresponding eigenvector components. We then merge nodes that lie on the same position (or very close to each other). This operation has the intrinsic property of preserving the relevant eigenvalues and the corresponding eigenvectors. In this sense it can be regarded as a decimation of the fast modes, without altering the slow modes, akin to k-space coarse graining, and eventually coming back to a real space coarse-grained network.

Spectral Coarse Graining represents an important shift with respect to most existing approaches, in particular clustering techniques, since it no longer aims at finding the "correct" communities of a network, which is often an ill-posed problem and requires heuristic methods. Instead of that, SCG ensures that the relevant properties of the network dynamics are preserved, which is the ultimate goal of any coarse graining strategy. Therefore it provides a quantitative way to approximate large networks with smaller ones. In the limit of exact SCG, the approximation becomes exact and only redundant information is removed from the network. In the perturbation approach, the validity of the approximation can be checked by evaluating the differences between eigenvalues or eigenvectors of the initial and the reduced network. Our results about stochastic matrices together with the work of [95] indicate that SCG is more accurate if eigenvalues are close to one.

The possibility of checking the coarse graining accuracy opens new ways to define an optimal coarse graining. A few directions have been outlined in this Thesis, but several other interesting issues emerge. Given the amount of precision required in the eigenvalues or eigenvectors, is there a way to find the absolute minimal number of groups? Reversely, given the number of groups, how should we form them in order to reach the highest accuracy in the eigenvalues? Should some node be split among different groups? More generally the goal of keeping some properties of the network leads naturally to the idea that groups could also be defined in order to preserve other quantities than eigenvalues or eigenvectors. For instance one could try to preserve percolation or epidemic thresholds. A particularly hot topic concerns synchronization in oscillatory networks since the coupling between nodes involves the Laplacian matrix. It has been shown in [12] that the condition for a linearly stable synchronized state in various kinds of oscillatory networks can often be expressed as $\frac{\lambda^1}{\lambda^{N-1}} < \beta$, where λ^1 and λ^{N-1} are the largest and smallest non-zero eigenvalues of L and β is a scalar depending on the system under scrutiny. Two interesting observations arise. First, under exact coarse graining the reduced network displays the same synchronization behavior as the initial one. Second, since λ_1 decreases and λ^{N-1} increases under coarse graining, we expect that, at a certain stage of the coarse graining, desynchronized networks will become synchronized. Remarkably, an optimal coarse graining preserving as much as possible the eigenvalues corresponds the slowest route towards the synchronized state, which is an interesting feature in several problems [7]. More generally, we suggest that focusing on the relevant eigenvectors is a good starting point for a coarse graining approach since global properties of the network are often reflected in the structure of these eigenvectors. In this respect, the choice of preserving eigenvalues and eigenvectors turns out to be very natural, first because they correspond to the slow modes (or Principle Components), and second because it has the advantage of leading straightforwardly to a simple graphical interpretation and to a well defined perturbation approach.

From a computational point of view we note that the time limiting step of the coarse

graining is the computation of the first eigenvectors. With the optimized routines available nowadays, networks of 10^4 nodes can be coarse grained within a few seconds since only the first eigenvectors are needed. However size remains an important hurdle when considering networks with $N > 10^5$. For these networks, one possibility is to use more advanced linear algebra routines combined with massive parallelization. Alternatively, a very elegant way would be to identify the groups so that the coarse graining preserves the most relevant spectral properties of the network, without having to compute eigenvectors. For instance in the exact coarse graining, only nodes with the same neighbors are grouped together. Thus groups can be formed considering the neighbors of each node and without calculating eigenvectors. In general this approach is restricted to the exact coarse graining and does not allow to significantly reduce the network size. However, considering some particular networks whose architecture obeys a deterministic process, it might be possible to infer the groups by simply considering the way the network is built. A recent study of the Configuration Space network of the Wako-Saitô-Muñoz-Eaton model [118] has already shown promising results in order to reach this goal [190].

To summarize, we have seen that spectral properties of networks are associated with several dynamical processes. In this Chapter, we have investigated two kinds of dynamics: random walks and Gaussian networks. By combining the use of Spectral Coarse Graining and Configuration Space Networks, we have shown that most dynamical systems can be analyzed as a random walk on a network. The example of di-alanine shows that the coarse graining performs remarkably well and leaves unchanged most properties of the dynamics. This finding allows to tackle the complexity of many dynamical systems by reducing the size of the corresponding configuration space. Considering the Laplacian L, we have seen that SCG allows to simplify proteins in the GNM framework, yet preserving the slow mode dynamics, which is the central feature of GNM. In the future it will be interesting to extend SCG to other kinds of dynamical processes on networks.

Chapter 8

General Conclusion

In this Thesis we have addressed the problem of simplifying complex systems of interacting units by reducing the complexity of the corresponding networks. From a theoretical point of view, this problem is indeed of high interest since complex networks or graphs are intricate mathematical objects on which standard approximation techniques cannot be applied in a straightforward way. From a practical point of view, reducing the complexity of networks by grouping nodes together yields a number of useful results. For instance it allows to classify nodes into natural groups which often reflect a common property of the nodes. In the case of dynamical systems on networks, reducing the complexity turns out to be often necessary since the large size of most real networks often hampers simulating dynamical processes involving global informations about the system. Finally from the point of view of visualization, it is extremely useful to find ways of reducing the network size as our eyes cannot cope with more than a few hundred nodes in the network.

Two different strategies have been investigated in this Thesis providing two different approaches to the problem of simplifying complex networks.

In the first Part we have dealt with the well-known clustering framework. Clustering methods are powerful tools to extract the structural organization of a network by grouping nodes into meaningful communities. Since the number of communities is much lower than the number of nodes, a reduced network of clusters is obtained in which nodes represent the functional or structural units of the original network. Clustering algorithms have found many applications in various fields of science and technology, such as informatics, social sciences, linguistics, biophysics, genetics, communication sciences, etc. The idea underlying the clustering approach is relatively easy to grasp since grouping nodes into communities is a natural way of organizing and labeling the units of the system under scrutiny. Unfortunately, from a mathematical point of view, it is much more delicate, and even cumbersome, to define what is meant by clusters or communities, so that the definition applies to any situation. Both the intuitive aspect of the community detection paradigm and the lack of proper definition have resulted in a very large amount of clustering algorithms available nowadays.

Rather than developing our personal algorithm, we have addressed the question of the reliability of the community structure. This is indeed a crucial issue since it has been observed that clustering algorithms applied on networks without any particular internal structure often partition the network into groups that do not account for real communities. In order to distinguish between effective community structure and artifacts of clustering algorithms, we have introduced the *clustering entropy*. Clustering entropy provides a measure of the stability of clusters when external noise is artificially introduced into the network. Moreover the use of external noise in the network allowed us to identify *unstable* nodes as the nodes flipping between clusters under different noisy realizations of the network. In the case of a synonymy network,

we have found that identifying unstable nodes was able to refine the automated classification of words as synonyms.

As another application, we have shown that clustering techniques combined with the detection of unstable nodes allows to unravel the large scale organization of a particularly interesting kind of networks describing dynamical systems: the Configuration Space Networks. In a Configuration Space Network each state of the dynamical system is associated with a node and each transition (either computed analytically or observed in a simulation) is associated with an edge. With this definition, the system dynamics is fully described by random walks on the network. In the case of Molecular Dynamics simulations of peptides, the network approach was shown to be particularly interesting since it does not require the projection of the underlying free-energy landscape onto one or two order parameters. In addition it allows to use the tools of complex networks to study the dynamics of the peptide and eventually to extract the large scale features of the free-energy landscape. Considering the network built from a MD simulation of a di-alanine peptide, we have shown that clusters identify correctly energy basins, and unstable nodes sample transition states.

At this point, a natural question arose: does a random walk on the network of clusters still represent the system dynamics? And if not, is there a way to reduce the complexity of the network without altering the random walk properties?

This question has been addressed in the second Part of this Thesis and more generally the possibility of reducing the network size and complexity while preserving some of its properties has been discussed. Our new method, which we refer to as *Spectral Coarse Graining* because of its similarity with the goals of coarse graining approaches used in statistical physics, differs from the usual clustering paradigm, though it is also based on the idea of grouping nodes. Rather than focusing on finding the "correct" communities in a network, it aims at preserving the relevant spectral properties of the original network in the coarse-grained version.

In the exact coarse graining, all non-zero eigenvalues and the corresponding eigenvectors are preserved, which implies that no information is removed from the system. More relevant for practical applications is the almost-exact coarse graining in which a few selected eigenvalues are left unchanged in the reduced network. Eigenvalues to preserve are naturally chosen as the ones corresponding to the slow modes, or equivalently to Principle Components. However, contrary to Principle Component Analysis in which only the dimensionality of a data set is reduced and not the size, Spectral Coarse Graining yields a smaller network that keeps as much information as possible from the initial one.

Spectral properties are often related to dynamical processes taking place on complex networks. In this Thesis two examples of dynamical processes have been considered: random walks and Gaussian networks. Random walks describe the dynamics observed in several kinds of networks, and are especially suited for Configuration Space Networks. Considering random walks on various real networks, we have shown that Spectral Coarse Graining leaves unchanged the large scale features of the random walk, while the network size was significantly decreased. In particular our method provides a consistent way of reducing the complexity of the space of configurations of dynamical systems by coarse graining the corresponding configuration space network without altering the dynamics. Finally we note that the example of di-alanine network illustrates remarkably the main difference between usual clustering and Spectral Coarse Graining. On the one hand clusters have been shown to represent energy basins, but random walks on the network of clusters do not represent the system dynamics. On the other hand the coarse-grained network was shown to preserve the system dynamics, but groups could not be straightforwardly associated with large-scale features of the free-energy landscape.

In the case of Gaussian networks describing the fluctuations from equilibrium position in proteins or polymers, we have been able to group nodes in a way that preserves the slow mode dynamics. We suggest that the coarse-grained protein may provide a good approximation of

the initial one, even considering more involved dynamics. Most often proteins have been coarse-grained according to their structure. To our knowledge this is the first time that a well-defined dynamical coarse graining scheme has been designed for proteins, ensuring that the slow mode dynamics is preserved.

In conclusion we believe that the two strategies described in this Thesis provide two complementary ways to tackle the complexity of systems represented as a network. On the one hand clustering techniques, despite their inherent ambiguities, are likely more appropriate for classifying the nodes of a network into meaningful communities. On the other hand Spectral Coarse Graining provides a new and well-defined way of approximating large networks by smaller ones and finds its natural framework considering dynamical processes on networks.

Acknowledgments

The present Thesis contains the different results of the research I performed at EPFL from February 2004 until July 2007 in the Laboratory of Statistical Biophysics under the supervision of Prof. Paolo De Los Rios. This work has been financially supported by COSIN (FET Open IST 2001-33555), DELIS (FET Open 001907) and the SER-Bern (02.0234).

I would like first to express my acknowledgments to Paolo De Los Rios. Thanks to his guidance throughout these 3 and half years, I could experience the pleasure of doing research on both theoretical and more applied problems. His availability and his patience have been a great help and our frequent discussions have been an invaluable source of inspiration for me.

I benefited a lot from the pleasant environment in the lab and I'd like to thank Thomas Petermann, Cecile Caretta Cartozo, Francesco Piazza, Mariangeles Serrano and Marco Zamparo for their nice and friendly attitude. I am particularly thankful to David Morton De La Chappelle who witnessed a high degree of motivation in joining and enlarging several projects I had been working on.

During my Thesis I had the opportunity to collaborate with Jean-Cédric Chappelier at EPFL and Amedeo Caflisch at University of Zürich and I'd like to thank them both for their availability and their coaching. I could not omit to mention my good friend and collaborator Francesco Rao with whom I enjoyed sharing ideas not only about research and science, but also about life and faith.

I am also indebted to Alessandro Flammini, Michele Vendruscolo and Felix Naef who accepted to be the three experts for this Thesis.

More personally I want express my recognition to my parents. Since my childhood you have developed my interests and stimulated my reflexions in various fields. Without your education and the trust you showed me I could not have reached this stage in my life. For this reason I am happy to dedicate to you this Thesis. Moreover I am extremely thankful to my brother and sisters for the affection they have witnessed and all the great times we have shared together.

Finally a special thank to Madeline Strinning who will become my wife in a few months. Your presence on my side helps me remembering every day what is really precious in life and I look forward spending this life with you.

Appendix A

Modeling Configuration Space Networks

Apart from community structure, other statistical measures have been used to understand the structure of configuration space networks. In an early study [148], CSN have been shown to exhibit, among else, a power-law degree distribution. The degree distribution has been extensively used to characterize network topology However for CSNs, the weight distribution is the most relevant and informative measure, since the weight of a node is proportional to the probability of visiting a configuration and therefore relates to the underlying free-energy landscape. In particular the weight does not depend on the saving time M, whereas the degree does.

Weight distribution has been studied in several other kinds of networks, providing a natural extension of the degree distribution to weighted networks [14, 188]. In this Appendix a complete discussion about the weight distribution and its physical interpretation is provided. Other topological features such as the degree distribution, the clustering coefficient and the average neighbor degree have been derived in [65] but are not included in the present manuscript.

A.1 Analytical derivation for the weight distribution

Here we show that the energy landscape ($U(\mathbf{x})$) and the weight distribution ($P(w)$) of the CSNs are related by an analytical formula. The weight of a node is defined as the number of times a configuration is visited during the simulation. In the continuous approximation and assuming the weight of each node is known exactly, $P_t(w)$ for $w > 0$ reads

$$P_t(w) = \frac{1}{V_t} \int_0^\infty dr \int r^{D-1} d\Omega \delta(w(r, \Omega, t) - w), \tag{A.1}$$

with V_t the volume of the space visited in the simulation and D the dimension. Ω is the solid angle in D-dimensional spherical coordinates and $w(r, \Omega, t)$ the weight of the node at position (r, Ω), at time t. For simplicity spherical symmetry of the energy landscape ($U(\mathbf{x}) = U(r)$) will assumed in the following.

Taking $U(r)$ in the units of $k_B T$, $U(0) = 0$ the expected weight of a node is proportional to the Boltzmann weight:

$$w(r, t) = w(0, t) \exp(-U(r)) \tag{A.2}$$

In first approximation, we assume that the weight of CSN nodes is given by its expected value of Eq. (A.2). In [93], it has been shown that an exact analytical derivation should include the fact that the weight of each node follows a binomial distribution, with a mean value given

by Eq. (A.2). However for large weights, our approximation is already excellent and changes are only observed for nodes with a very low weight, which are typically not crucial to infer the behavior of the weight distribution. In addition, using Eq. (A.2) keeps the analytical derivation as simple as possible.
The properties of $\delta(f(x))$ give [1]:

$$\delta(w(r,t)-w) = \delta(w(0,t)\exp(-U(r)) - w) = \sum_{i=1}^{n} \frac{\delta(r-r_i^*)}{|wU'(r_i^*)|}$$

r^* is given by the implicit equation:

$$\exp(-U(r_i^*)) = \frac{w}{w(0,t)}, \qquad i = 1...n \tag{A.3}$$

where n is the number of simple zeros of $w(r,t) - w$ (values of w such that $w(r,t) - w$ has no zero are excluded and for such w, $P_t(w) = 0$). If the energy landscape is a monotonously increasing function along r, $n = 1$ for any w. Inserting Eq. (A.3) into Eq. (A.1) gives [64]:

$$\begin{aligned} P_t(w) &= C_t \frac{1}{w} \int_0^\infty dr\, r^{D-1} \sum_{i=1}^{n} \frac{\delta(r-r_i^*)}{|U'(r_i^*)|} \\ &= \frac{C_t}{w} \sum_{i=1}^{n} \frac{{r_i^*}^{D-1}}{|U'(r_i^*)|} \end{aligned} \tag{A.4}$$

with C_t the appropriate normalizing factor. In the following the normalization is always done and the time dependence will be dropped for simplicity.

The first important remark concerns the w^{-1} factor in the weight distribution. This factor does not depend on the particular shape of the energy landscape, neither on the dimension D. Thus we expect any weight distribution to have a power-law, $P(w) \propto w^{-\gamma}, \gamma = 1$, multiplied by a correcting factor which depends on the energy landscape. Although the correcting factor can be anything, it often turns out to be a logarithmic correction.

A similar derivation has been performed by Krivov *et al.* in [93] under the hypothesis that $\rho(U) \propto U^\gamma$, where $\rho(U)$ is the density of configurations with energy U. Starting from $P(w)\mathrm{d}w = \Omega(U)\mathrm{d}U$ and using that $U(\mathbf{x}) = -\log(w/w_0) \Leftrightarrow \mathrm{d}U = -\mathrm{d}w/w$, they found the following expression:

$$\Leftrightarrow P(w) \propto \frac{1}{w} \left(\log(w_0/w)\right)^\gamma \tag{A.5}$$

Eq. (A.4) is useful when considering any kind of energy landscape with spherical symmetry. Even if r^* cannot be computed analytically, the equation can be always solved numerically. On the other hand Eq. (A.5) is useful to treat a special kind of energy landscape satisfying $\rho(U) \propto U^\gamma$, with the draw-back that this condition is not always easily verified.

A.2 Simple Energy Landscape Models

In general free-energy landscapes of real systems are extremely complex, such that even writing down a mathematical expression is often impossible. However close to the minimum of an enthalpic basin, i.e. for nodes with a large weight, such systems can often be approximated by means of a Taylor expansion of the energy landscape, whose first term is harmonic. It is

[1] For a given function $f(x)$ with n simple zeros ($f(x_i^*) = 0, f'(x_i^*) \neq 0, i = 1...n$), we have:

$$\delta(f(x)) = \sum_{i=1}^{n} \frac{\delta(x-x_i^*)}{|f'(x_i^*)|}$$

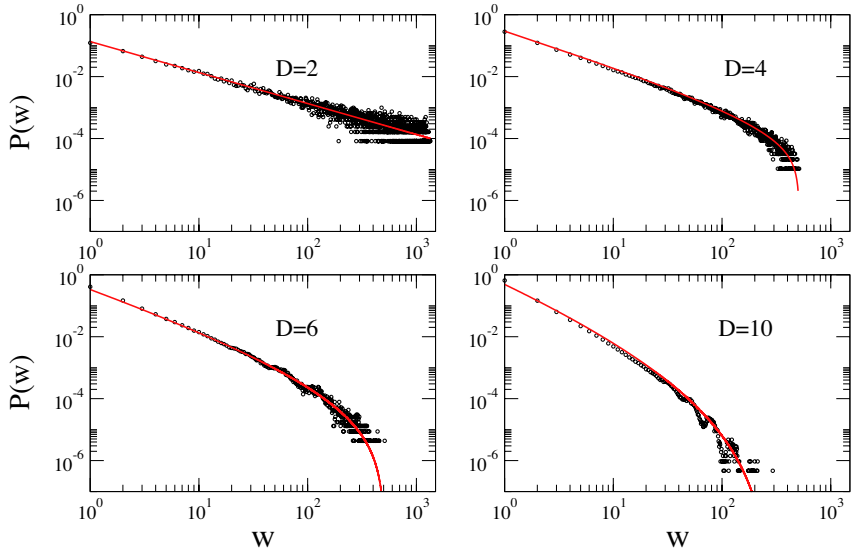


Figure A.1: Weight distribution for the quadratic well energy landscape ($\alpha = 5$) in four different dimensions. Circles ($\circ$) account for the weight distributions observed in CSNs built from the potential of Eq. (A.6). The red curve shows the analytical estimate with $w(0)$ taken as the weight of the heaviest node and C the appropriate normalization. Parameters are: $D = 2$, $a = 0.02$, $N_s = 2 \cdot 10^6$; $D = 4$, $a = 0.1$, $N_s = 2 \cdot 10^6$; $D = 6$, $a = 0.2$, $N_s = 2 \cdot 10^6$; $D = 10$, $a = 0.3$, $N_s = 5 \cdot 10^6$. a is the distance between two neighbor sites in the discretization, N_s is the number of snapshots.

therefore expected that lots of insights in the weight distribution of real CSNs, and especially in its large weight behavior, can be obtained by simplified energy models.

In this Section the weight distribution for simple energy landscapes is derived from Eq. A.4 and confronted to CSN topology, showing that the heavy tail behavior can be understood as an effect of the enthalpic nature of the basins. CSNs have been build as described in Chapter 5.

A.2.1 The quadratic well

The quadratic well in spherical coordinates is given by:

$$U(r) = \alpha r^2 \qquad\qquad \alpha > 0 \tag{A.6}$$

Because of the simple form of the potential, r^* of Eq. (A.3) is easily computed:

$$r^* = \left(-\frac{1}{\alpha} \ln(\frac{w}{w(0)}) \right)^{\frac{1}{2}},$$

which gives the following weight distribution:

$$P(w) = C \frac{1}{w} \left[\ln \left(\frac{w(0)}{w} \right) \right]^{\frac{D}{2}-1}. \tag{A.7}$$

The weight distribution $P(w)$ follows a power-law of exponent -1 with a logarithmic correction for $D > 2$. This behavior is verified numerically in Figure A.1. Black circles represent

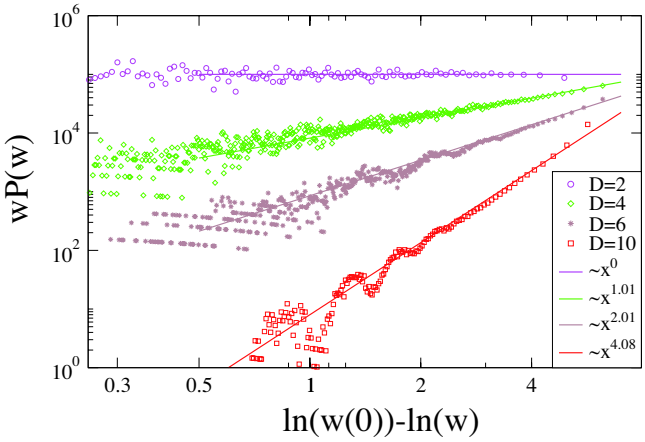


Figure A.2: Rescaled weight distribution. For $D = 2, 4, 6, 10$, the slopes computed by linear regressions are $0, 1.01, 2.01, 4.08$, respectively, and thus are close to the expected values. Note that for large weights the fluctuations due to discretization become important and the data do not follow anymore a straight line. For clarity, the data for $D = 2$ have been shifted in order not to overlap.

the topology of CSNs built from MC simulations on the potential of Eq. (A.6). Continuous lines have been obtained from Eq. (A.7) with the fitting parameters $w(0)$ as the weight of the heaviest node visited during the simulation and C the appropriate normalization. The presence of apparent waves in D=10 stems from the discretization of the energy landscape. Actually the distribution is composed of a sum of Poisson distributions corresponding to each level of energy allowed in the discredited system and the analytical curve describes in first approximation the envelope of these distributions.

As predicted, the logarithmic correction becomes more and more significant as D increases, such that some of the distributions seem to follow a power-law with an exponent smaller than -1. A careful inspection of the case $D = 10$ shows that the analytical prediction is not completely correct for low weight nodes. This is due to the approximation of considering the weight of each node as equal to its expected value, as it was already pointed out in the previous Section. However, the analytical curves describe well the general behavior observed in simulations. Figure A.2 displays a rescaling of $P(w)$ where straight lines are linear regressions. From Eq. (A.7) the slopes should be equal to $D/2 - 1$. As seen in Figure A.2 results for CSNs weight distribution are very close to the analytical ones. Though this rescaling seems appealing to extract the effective dimension of a system, we observed that the slopes are quite sensitive to finite sampling, especially for large w. Therefore this analysis should be used with great care to infer the dimensionality of a system.

The quadratic well is typically a model of enthalpic basin with a single minimum and might not be relevant for all kinds of energy landscapes. In particular it has been recently suggested that several energy basins are characterized by a large number of equivalently populated configurations [91]. Such ensembles are referred to as entropic basins. As a model for entropic basins, the square well is studied in the next Section.

A.2.2 The square well

The square well with spherical symmetry is defined by the following equation

$$U(r) = \begin{cases} 0 & \text{if } r \leq 1; \\ \infty & \text{if } r > 1. \end{cases} \tag{A.8}$$

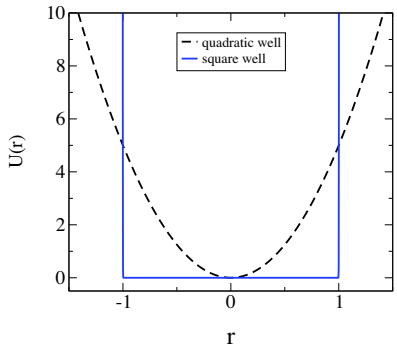


Figure A.3: The quadratic (dotted line) and square well (blue line) energy functions in $D = 1$.

Contrary to the quadratic well (see Figure A.3 for comparison), all the sites with $r \leq 1$ have an equal probability of being visited. This leads to a flat stationary solution for $w(r)$ and a delta function for $P(w)$. In real CSNs built from Eq. (A.8) a Poisson distribution is obtained, due to finite size effects. The Poisson distribution has a mean value at $\bar{w} = 255$, corresponding the average weight, i.e the number of snapshots ($N_s = 2 \cdot 10^6$) divided by the number of possible sites: $\pi \cdot (1/a)^2 \approx 7850$, $\Rightarrow \bar{w}_{exp} = 255$, where $a = 0.02$ is the distance between two neighbor sites in the lattice defined in the discretization process. This distribution completely differs from the one obtained with a quadratic potential, as illustrated in Figure A.3, reflecting the different nature of the two energy landscapes.

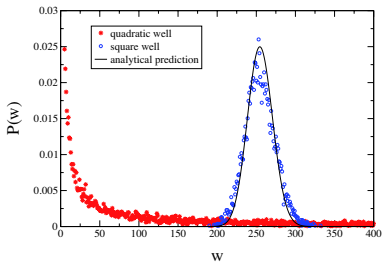


Figure A.4: Comparison between the weight distribution of the square well and the quadratic well in $D = 2$ dimensions. We have used $a = 0.02$ and $N = 2 \cdot 10^6$ in both cases to run the Monte Carlo simulation. The distribution for the square well follows a Poisson distribution (continuous line).

A.2.3 The Mexican Hat

The two cases studied above illustrate two different types of energy landscapes. In the quadratic well, the system dynamics is mainly driven by the potential gradient, and the energy basin is *enthalpic*, whereas for the square well, the energy basin is *entropic*. It is important to note that $P(w)$ bears the signature of the difference between these two cases. An example containing both kinds of energy landscapes, i.e., entropic and enthalpic, is the "Mexican-Hat" landscape model (see Figure A.5A), already studied in Chapter 5. The potential is defined by

$$U(r) = 40(r^6 - 1.95r^4 + r^2)$$

In $D > 1$ dimensions, the central basin is enthalpic and has a minimum at $r = 0$, while the surrounding basin is entropic and forms a shell centered at $r = 0.97$. On a log-log plot, $P(w)$ of the central basin ($r < \hat{r}$) has a broad tail, which is typical for an enthalpic well (Figure A.5B). On the other hand, $P(w)$ of the surrounding basin ($r > \hat{r}$) shows a rather flat region followed by a sharp decay. This decay is typical of an entropic basin as the square well. A pure Poisson shape is not obtained since the basin has an enthalpic component along r. As it can be seen in Figure A.5C, low weight nodes are located either close to $\hat{r}$ or have $r > 1.05$. The nodes in the surrounding basin with a large weight are found close to the minimum r^*. Finally the heaviest node of the network is at $r = 0$, as expected.

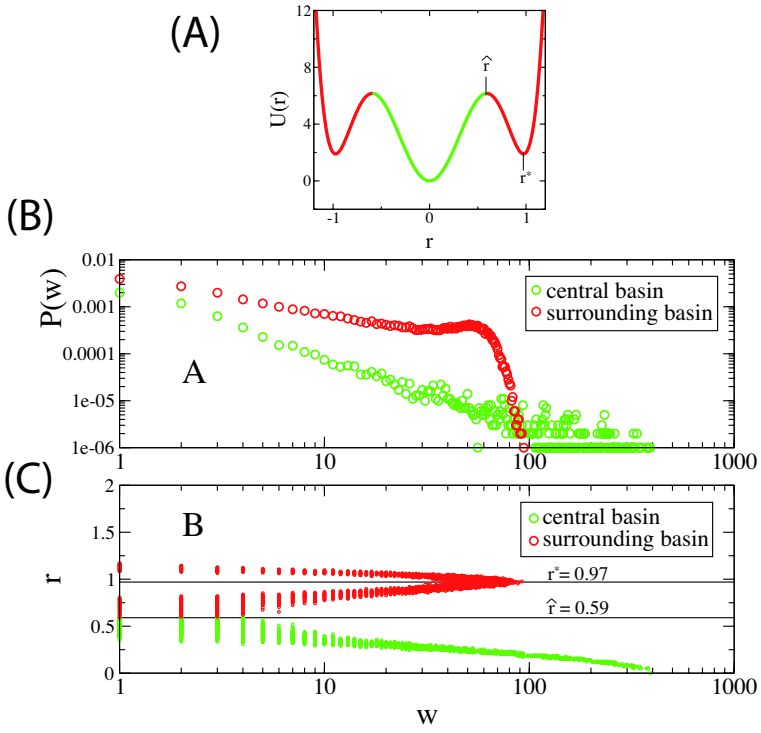


Figure A.5: **(A)** Energy function of the Mexican-Hat model along the radial coordinate r. **(B)** Weight distribution for the two basins of the Mexican-Hat model in $D = 3$ dimensions. The nodes have been attributed to the basins according to the coordinate r. **(C)** Radial coordinate r of every node as a function its weight. $r = 0.97$ corresponds to the minimum in the surrounding basin (entropic). $r = 0.59$ corresponds to the maximum separating the two basins. $a = 0.05$, $N = 10^6$.

A.3 Di-alanine weight distribution

We have seen in Chapter 5 that the CSN of di-alanine consists in 4 main clusters, corresponding to 4 energy basins. The weight distributions of the four clusters of Figure 5.7 are displayed in Figure A.6A. The distributions follow a power-law $w^{-\delta}$ with exponent $\delta \approx 1$ that is consistent with the behavior predicted by Eq. (A.7). Results presented in Figure A.6 have been obtained with a logarithmic binning. Changing the bin size did not alter the slope of the weight distribution (data not shown). These findings indicate that indeed the behavior of the weight distribution of CSNs can be explained by simple energy model. In the case of di-alanine, the

energy basins are known to be mostly enthalpic. As a matter of fact the weight distribution follows the one predicted by the quadratic well approximation. Moreover the slope of the weight distribution is very robust against the changes in the size of the discretization cells, as shown in Figure A.6B.

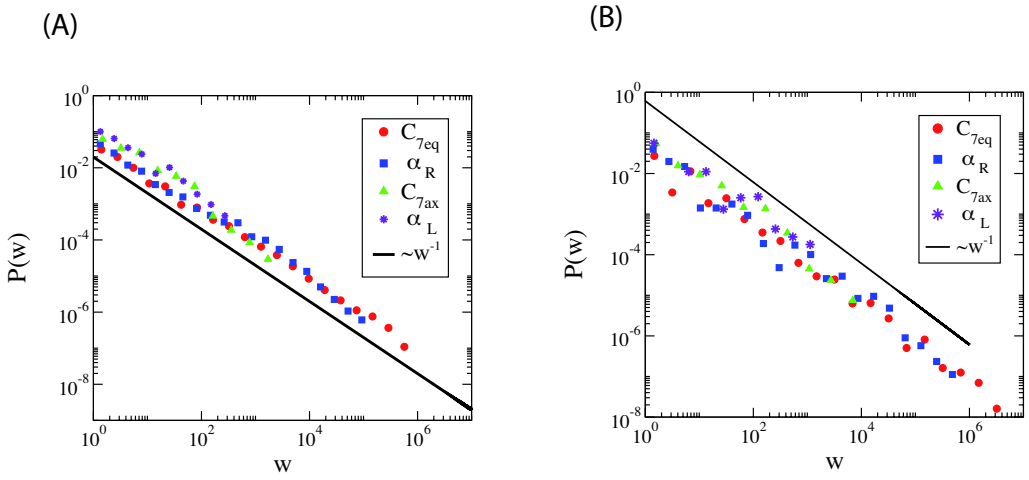


Figure A.6: Weight distribution of the four clusters in the di-alanine CSN considering the dihedral angle discretization. **(A)** 50×50 discretization of the (ϕ,ψ) space. **(B)** 20×20 discretization of the (ϕ,ψ) space. Each basin is represented by a different color and the solid line w^{-1} is shown to guide the eyes. Colors are the same as in Figures 5.7 and 5.8.

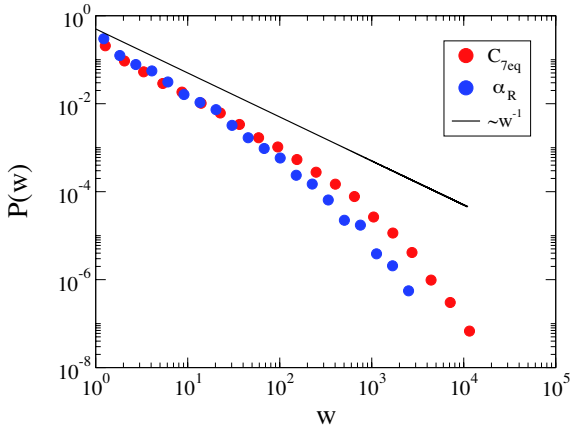


Figure A.7: Weight distribution of the two main clusters in the di-alanine CSN considering the inter-atomic distance discretization (Figure 5.9 of Chapter 5).

Another discretization procedure based on the atomic distances has also been carried out and yielded a CSN whose community structure was close to the one obtained considering the dihedral angles (see Figure 5.9 in the main text). Using atomic distances introduces a larger number of parameters than the only two dihedral angles. Hence the weight distribution is not expected to behave exactly in the same way as for dihedral discretization. The analytical results for the quadratic well suggest even that it should decay faster, in particular for nodes with a large weight. This behavior is observed in Figure A.7. The weight distribution of the two larger communities represented with red and blue colors in Figure 5.9 exhibits clearly a decay faster

than w^{-1}. This discrepancy may appear contradictory since the same dynamical system yields two different network topologies. However, we have to remember that any discretization scheme corresponds to a projection on some (maybe many) degrees of freedom of the system. As more parameters are considered, the weight distribution appears more "high-dimension-like". This feature was already present in the example of the quadratic well. In $D = 3$ dimensions, the weight of a node $w(x, y, z) \propto \exp\left(-\frac{1}{2}(x^2 + y^2 + z^2)\right)$. Now if only x and y are considered as degrees of freedom, the weight of the nodes is $w(x, y) \propto \int \mathrm{d}z \exp\left(-\frac{1}{2}(x^2 + y^2 + z^2)\right) \propto \exp\left(-\frac{1}{2}(x^2 + y^2)\right)$, which results in a weight distribution equivalent to the $D = 2$ case. More generally using a reduced set of degrees of freedom is equivalent to decreasing the apparent dimension observed in the weight distribution.

Appendix B

Pathological eigenvalues in Spectral Coarse Graining

B.1 Exact coarse graining: second case

The central idea underlying the exact coarse graining procedure described in Section 7.2 of part II is that two nodes having the same neighbors have equal components along their left eigenvectors $\langle u^\alpha|$ for $\lambda^\alpha \neq 0$ and can be merged without altering the spectral properties of the stochastic matrix W.

In this appendix we address the question whether there exist situations in which $u_1^\alpha = u_2^\alpha$, while the nodes do not have the same neighbors.

Of particular interest is the case in which two nodes have the same neighbors plus are connected to each other (Fig. B.1). As in Fig. 7.1 it is natural to coalesce such pair of nodes, although they do not have *exactly* the same neighbors..

In general if two nodes with degree n have $n-1$ neighbors in common and are connected to each other, the stochastic matrix reads:

$$W = \begin{pmatrix} 0 & \frac{1}{n} & \dots \\ \frac{1}{n} & 0 & \dots \\ W_1^T & W_2^T & \vdots \end{pmatrix}$$

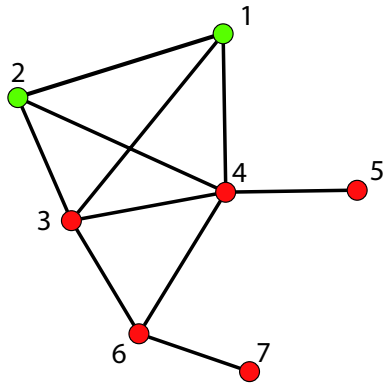


Figure B.1: Small graph presenting a second ideal case for the SCG of networks. The green nodes have exactly the same neighbors except that they are connected to each other.

with $W_1 = (W_{31}, ..., W_{N1})$ and $W_2 = (W_{32}, ..., W_{N2})$. Since nodes 1 and 2 share all their neighbors except themselves, $W_1 = W_2$.

We first show that W has an eigenvalue equal to $\frac{-1}{n}$. The eigenvalues of W satisfy the equation $\det(\lambda I - W) = 0$. Since subtracting one column of the matrix to another one does not change the determinant, we can subtract column 2 from column 1 in matrix $\lambda I - W$. After this operation only the two first entries of column 1 are non-zero, with the first one equal to $\lambda + \frac{1}{n}$ and the second one equal to $-\lambda - \frac{1}{n}$. We can conclude that $\lambda = \frac{1}{n}$ is an eigenvalue of W.

Let us consider now an eigenvector $\langle u^\alpha|$. $\langle u^\alpha|$ satisfies the two following equations:

$$\begin{cases} u_2^\alpha \frac{1}{n} + \sum_{i=3}^N W_{i1} u_i^\alpha &= \lambda^\alpha u_1 \\ u_1^\alpha \frac{1}{n} + \sum_{i=3}^N W_{i2} u_i^\alpha &= \lambda^\alpha u_2 \end{cases}$$

$$\Leftrightarrow (u_1^\alpha - u_2^\alpha)\frac{1}{n} = -\lambda(u_1^\alpha - u_2^\alpha) \tag{B.1}$$

Eq. (B.1) implies that either $u_1^\alpha = u_2^\alpha$ or $\lambda = -\frac{1}{n}$. Moreover the eigenvalue $-\frac{1}{n}$ is no longer present in the reduced matrix. Therefore merging the two nodes has the effect of removing from the spectrum one eigenvalue equal to $-\frac{1}{n}$.

This result can be extended to the case in which m nodes with degree n form a clique and share all their other $n-(m-1)$ neighbors. By suitably subtracting the columns of $\lambda I - W$, one immediately shows that $\lambda = -\frac{1}{n}$ is an eigenvalue with multiplicity $m-1$. Similar arguments than the one leading to Eq. (B.1) allow to conclude that $u_i^\alpha = u_j^\alpha$ for all nodes in the clique if $\lambda^\alpha \neq -\frac{1}{n}$.

As a final remark we point out that the ideal case of Section 7.2 could be immediately satisfied in Fig. B.1 by adding self loops to all nodes. However, the nodes of Fig. 7.1 in the main text would not have any longer the same neighbors. An argument similar to the one presented in this Appendix shows that if two nodes not connected between each other and with degree n have $n-1$ common neighbors plus a self-loop, an eigenvalue equal to $\frac{1}{n}$ appears in the spectrum and the condition $u_1^\alpha = u_2^\alpha$ is satisfied only for $\lambda^\alpha \neq \frac{1}{n}$.

Apart from a better mathematical understanding, those results have a crucial consequence for SCG. If we want to ensure that nodes are correctly merged in the two ideal cases of Fig. 7.1 and B.1, we should not consider eigenvectors $\langle u^\alpha|$ such that λ^α is equal to 0, $\frac{1}{n}$ or $\frac{-1}{n}$, $\forall n \in \mathbb{N}$.

Nevertheless the interesting features that should be preserved under coarse graining are most often contained in the first eigenvalues, which always have values significantly larger than $\frac{1}{2}$ for large networks. Hence the risk of considering by bad luck $\langle u^\alpha|$ with $\lambda^\alpha = \frac{1}{n}$ in SCG is relatively low.

B.2 Coarse graining the Laplacian matrix

In this section we discuss some peculiarities arising when applying Spectral Coarse Graining defined in Section 7.8.1 on the Laplacian matrix. We first stress that the presence of $D_{ii} = \sum_j A_{ij}$ on the diagonal changes the spectral properties of L compared to A, and thereby the coarse graining.

In particular the situation in which two nodes 1 and 2 have exactly the same neighbors does not result in two identical lines in L. Instead we have $L_{11} = L_{22}$, $L_{12} = L_{21}$ and $L_{1i} = L_{2i}$ $\forall i > 2$. In terms of an eigenvector v^α the following equations, among others, are to be satisfied:

$$\begin{cases} L_{11} v_1^\alpha + L_{12} v_2^\alpha + \ldots &= \lambda^\alpha v_1^\alpha \\ L_{21} v_1^\alpha + L_{22} v_2^\alpha + \ldots &= \lambda^\alpha v_2^\alpha \end{cases}$$

Subtracting the tow lines leads to:

$$(v_1^\alpha - v_2^\alpha)(L_{11} + L_{12}) = \lambda^\alpha (v_1^\alpha - v_2^\alpha)$$

implying that either $v_1^\alpha = v_2^\alpha$ or $\lambda^\alpha = L_{11} + L_{12} = \hat{k_1}$, where $\hat{k_1}$ is equal to the reduced degree of node 1, excluding the possible self-edge or edge to node 2. Using the same argument as before (see Section B.1), one shows that eigenvalue $\lambda^\beta = \hat{k_1}$ always exists. The eigenvector $|v^\beta\rangle$ has several important properties which can be derived from the orthonormality of $\{|v^\alpha\rangle\}$. The subspace defined by $\{|v^1\rangle, ...|v^{\beta-1}\rangle, |v^{\beta+1}\rangle, ...|v^N\rangle$ consist of all possible vectors $|v\rangle$ with $v_1 = v_2$. Since $|v^\beta\rangle$ is perpendicular to this subspace, we have that $v_1^\beta = -v_2^\beta$ and $v_i^\beta = 0 \; \forall i > 2$.

To summarize, two nodes having exactly the same neighbors lead to a particular eigenvector with opposite components for these two nodes. As a consequence, such eigenvector should not be considered in the exact SCG procedure, as it was the case for eigenvectors of W corresponding to zero eigenvalues. For simple networks, the pathological eigenvalues L are easily identified since they correspond to integers. Fortunately, this situation is not often encountered since relevant eigenvalues of L are close to 0 and much lower than 1 for large networks.

From the perspective of information encoded in each eigenvector, λ^β appears explicitly in the spectral decomposition of L, which is somehow in contradiction with an exact coarse graining removing only the redundant information of the network. However, writing $L_{ij} = \sum_\alpha \lambda^\alpha v_i^\alpha v_j^\alpha = \sum_\alpha \lambda^\alpha L_{ij}^\alpha$ shows that λ^β only appears in $L_{11}^\beta = L_{22}^\beta = \lambda^\beta (v_1^\beta)^2, L_{12}^\beta = L_{21}^\beta = -\lambda^\beta (v_1^\beta)^2$. Moreover this contribution completely vanishes when coarse graining the network, i.e. $\tilde{L_{11}}^\beta = \frac{1}{2}(L_{11}^\beta + L_{12}^\beta + L_{21}^\beta + L_{22}^\beta) = 0$. Therefore the pathological eigenvalue λ^β naturally disappears under SCG.

The previous derivation generalizes straightforwardly to groups including more than two nodes and shows that contrary to the stochastic matrix, the redundancy in the Laplacian is not encoded in null eigenvalues. Instead some particular eigenvalues account only for elements of L corresponding to edges between equivalent nodes. Such edges do not play any role in the coarse-grained network and do not appear in the reduced Laplacian $\tilde{L}$, as did loops in L. It is therefore perfectly sounded to speak about exact coarse graining, though the redundancy is not contained in 0 eigenvalues.

Bibliography

[1] H. Agrawal and E. Domany. Potts ferromagnets on coexpressed gene networks: Identifying maximally stable partitions. *Phys. Rev. Lett.*, 90(25):158102, 2003.

[2] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74(1):47–97, 2002.

[3] I. Alvarez-Hamelin, L. Dall'Asta, A. Barrat, and A. Vespignani. k-core decomposition: a tool for the analysis of large scale internet graphs. *cs.NI/0511007*, 2005.

[4] M. Andrec, A. K. Felts, E. Gallicchio, and R. M. Levy. Chemical Theory and Computation Special Feature: Protein folding pathways from replica exchange simulations and a kinetic network model. *Proceedings of the National Academy of Science*, 102:6801–6806, May 2005.

[5] C. B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181:223–230, 1973.

[6] J. Apostolakis, P. Ferrera, and A. Caflisch. Calculation of conformational transitions and barriers in solvated systems: Application to the alanine dipeptide in water. *Journal of Chemical Physics*, 110(4):2099–2108, 1999.

[7] A. Arenas, A. Díaz-Guilera, and C. J. Pérez-Vicente. Synchronization reveals topological scales in complex networks. *Phys. Rev. Lett.*, 96:114102, 2006.

[8] A. R. Atilgan, S. R. Durrell, R. L. Jernigan, M. C. Demirel, O. Keskin, and I. Bahar. Anisotropy of fluctuation dynamics of proteins with an elastic network model. *Biophysical Journal*, 80:505–515, 2001.

[9] I. Bahar, A. R. Atilgan, M. C. Demirel, and B. Erman. Vibrational dynamics of folded proteins: Significance of slow and fast motions in relation to function and stability. *Phys. Rev. Lett.*, 80(12):2733–2736, Mar 1998.

[10] J. R. Banavar, A. Flammini, D. Marenduzzo, A. Maritan, and A. Trovato. Geometry of compact tubes and protein structures. *ComPlexUs*, 1:4–13, 2003.

[11] A-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[12] M. Barahona and L. M. Pecora. Synchronization in small-world systems. *Phys. Rev. Lett.*, 89:054101, 2002.

[13] A. Baronchelli and V. Loreto. Ring structures and mean first passage time in networks. *Phys. Rev. E*, 73:026103, 2006.

[14] A. Barrat, M. Barthélemy, R. Pastor-Satorras, and A. Vespignani. The architecture of complex weighted networks. *PNAS*, 101(11):3747–3752, 2004.

[15] A. Barrat, M. Barthélemy, and A. Vespignani. Weighted evolving networks: Coupling topology and weight dynamics. *Phys. Rev. Lett.*, 92:228701, 2004.

[16] R.B. Best and G. Hummer. Reaction coordinates and rates from transition paths. *Proc. Natl. Acad. Sci. USA*, 102:6732–6737, 2005.

[17] M. Blatt, S. Wiseman, and E. Domany. Superparamgnetic clustering of data. *Phys. Rev. Lett.*, 76(18):3251–3254, 1996.

[18] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang. Complex networks: Structure and dynamics. *Pysics Report*, 424:175–308, 2006.

[19] P G Bolhuis, C Dellago, and D Chandler. Reaction coordinates of biomolecular isomerization. *Proc Natl Acad Sci U S A*, 97:5877–5882, 2000.

[20] B. Bollobás. The evoultion of sparse graphs. In *Graph Theory and Combinatorics*, pages 35–37, London, 1984. Academic Press.

[21] B. Bollobás. *Random graphs*. Academic Presse, London, 1985.

[22] S. Bornholdt and H. G. Schuster. *Handbook of Graphs and Networks: From the Genome to the Internet*. Wiley-VCH, 2002.

[23] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25(2):163–177, 2001.

[24] S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proc. 7th International WWW Conference*, pages 107–117, 1998.

[25] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. Charmm: a program for macromolecular energy minimization and dynamics calculation. *Journal of Computational Chemistry*, 4:187–217, 1983.

[26] A. Caflisch. Network and graph analyses of folding free energy surfaces. *Current Opinion in Structural Biology*, 16:71–78, 2006.

[27] G. Caldarelli, A. Capocci, P. De Los Rios, and M. A. Muñoz. Scale-free networks from varying vertex intrinsic fitness. *Phys. Rev. Lett.*, 89:258702, 2002.

[28] A. Capocci, V. D. P. Servedio, G. Caldarelli, and F. Colaiori. Communities detection in large networks. *Lecture Notes in Computer Science*, 3243:181–187, 2004.

[29] C. Caretta Cartozo, P. De Los Rios, F. Piazza, and P. Lío. Bottleneck genes and community structure in the cell cycle network of s. pombe. *PLoS comupational biology*, 2007.

[30] M. Catanzaro, Caldarelli. G., and L. Pietronero. Social network growth with assortative mixing. *Physica A: Statistical Mechanics and its Applications*, 338:119–124, 2004.

[31] C. Chennubhotla and I. Bahar. Markov methods for hierarchical coarse-graining of large protein dynamics. *Lecture Notes in Computer Science*, 3909:379–393, 2006.

[32] R. J. Cho, M. J. Huang, M.and Campbell, H. Dong, L. Steinmetz, L. Sapinoso, G. Hampton, S. J. Elledge, R. W. Davis, and D. J. Lockhart. Transcriptional regulation and function during the human cell cycle. *Nature Genetics*, 27:48–54, 2001.

[33] A. Clauset, M. E. J. Newman, and C. Moore. Finding community structure in very large networks. *Phys. Rev. E*, 70:066111, 2004.

[34] R. J. Creswick, H. A. Farach, and C. P. Poole. *Introduction to renormalization group methods in physics*. John Wiley & Sons. Inc., 1992.

[35] Payel Das, Mark Moll, Hernan Stamati, Lydia E Kavraki, and Cecilia Clementi. Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction. *Proc Natl Acad Sci U S A*, 103:9885–90, 2006.

[36] I. Derényi, G. Palla, and T. Viscek. Clique percolation in random networks. *Phys. Rev. Lett.*, 94:160202, 2005.

[37] C.M. Dobson and M. Karplus. The fundamentals of protein folding: Bringing together theory and experiment. *Curr. Opin. Struct. Biol.*, 9:92–102, 1999.

[38] L. Donetti and M. A. Muñoz. Detecting networks communities: a new systematic and efficient algorithm. *J. Stat. Mech.*, page 10012, 2004.

[39] S. N. Dorogovstev, A. V. Goltsev, and J. F. F. Mendes. k-core organization of complex networks. *Phys. Rev. Lett.*, 96:040601, 2006.

[40] S. N. Dorogovtsev and J. F. F. Mendes. Evolution of networks. *Advances in Physics*, 51(4):1079–1187, 2002.

[41] S. N. Dorogovtsev and J. F. F. Mendes. *Evolution of networks: from biological nets to the internet and the WWW*. Oxford University Press, 2003.

[42] J. P. K. Doye. Network topology of a potential energy landscape: A static scale-free network. *Phys. Rev. Lett.*, 88(23):238701, 2002.

[43] R. Du, V. S. Pande, A. Y. Grosberg, T. Tanaka, and E. Shakhnovich. On the transition coordinate for protein folding. *J. Chem. Phys.*, 108:334–350, 1998.

[44] J. Duch and A. Arenas. Community detection in complex networks using extremal optimization. *Phys. Rev. E*, 72:027104, 2005.

[45] J-P. Eckmann and Moses E. Curvature of co-links uncovers hidden thematic layers in the world wide web. *Proc. Nat. Acad. Sci.*, 99(9):5825, 2002.

[46] V. M. Eguíluz, D. R. Chialvo, G. A. Cecchi, M. Baliki, and A. V. Apkarian. Scale-free brain functional networks. *Phys. Rev. Lett.*, 94:018102, 2005.

[47] A. J. Enright, S. Van Dongen, and C. A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Research*, 30:1575–1584, 2002.

[48] P. Erdös and A. Rényi. On random graphs. *Publ. Math. Debrecen*, 6:290–297, 1959.

[49] K. A. Eriksen, I. Simonsen, S. Maslov, and K. Sneppen. Modularity and extreme edge of the internet. *Phys. Rev. Lett.*, 90(14):148701, 2003.

[50] M. Fiedler. Algebraic connectivity of graphs. *Czechooslovak Math. Journal*, 23:298–305, 1973.

[51] M. Fiedler. A property of eigenvectors of non-negative symmetric matrices and its application to graph theory. *Czechoslovak Math. Journal*, 25:619–633, 1975.

[52] P.J. Flory. Statistical thermodynamics of random networks. *Proc. Roy. Soc. Lond. A.*, 351:351–380, 1976.

[53] S. Fortunato and M. Barthélémy. Resolution limit in community detection. *Proc. Natl. Acad. Sci.*, 104:36–41, 2007.

[54] S. Fortunato, A. Flammini, and F. Menczer. Scale-free network growth by ranking. *Phys. Rev. Lett.*, 96:218701, 2006.

[55] S. Fortunato, V. Latora, and M. Marchiori. Method to find community structures based on information centrality. *PRE*, 70:056104, 2004.

[56] H. Frauenfelder, S.G. Sligar, and P.G. Wolynes. The energy landscapes and motions of proteins. *Science*, 254:1598–1603, Dec 1991.

[57] L. Freeman. *Sociometry*, 40:35, 1977.

[58] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[59] D. Garlaschelli and M. I. Loffred. Maximum likelihood: extracting unbiased information from complex networks. *arXiv:cond-mat/0609015*, 2006.

[60] B. Gaveau and L. S. Schulman. Dynamical distance: coarse grains, pattern recognition, and network analysis. *Bulletin des Sciences Mathématiques*, 129:631–642, 2005.

[61] D. Gfeller, J-C. Chappelier, and P. De Los Rios. Finding instabilities in the community structure of complex networks. *Phys. Rev. E*, 72:056135, 2005.

[62] D. Gfeller, J.-C. Chappelier, and P. De Los Rios. Synonym dictionary improvement through markov clustering and clustering stability. In *Proc. of Int. Symp. on Applied Stochastic Models and Data Analysis (ASMDA'05)*, 2005.

[63] D. Gfeller and P. De Los Rios. Spectral coarse-graining of complex networks. *Phys. Rev. Lett.*, in press, 2007.

[64] D. Gfeller, P. De Los Rios, A. Caflisch, and F. Rao. Complex networks analysis of free-energy landscapes. *Proc. Natl. Acad. Sci.*, 104:1817–1822, 2007.

[65] D. Gfeller, P. De Los Rios, D. Morton De La Chapelle, G. Caldarelli, and F. Rao. Uncovering the topology of configuration space networks. *Phys. Rev. E*, submitted, 2007.

[66] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–7826, 2002.

[67] N. Go and H. Abe. Noninteracting local-structure model of folding and unfolding transition in globular proteins. *Biopolymers*, 20:911–1011, 1981.

[68] K.-I. Goh, G. Salvi, B. Kahng, and D. Kim. Skeleton and fractal scaling in complex networks. *Phys. Rev. Lett.*, 96:018701, 2006.

[69] I. Goldhirsch and Y Gefen. Analytic method for calculating properties of random walks on networks. *Phys. Rev. A*, 33(4):2583–2594, 1986.

[70] M. Goldstein. Viscous liquids and glass transition - a potential energy barrier picture. *Journal Of Chemical Physics*, 51:3728, 1969.

[71] G. H. Golub and C. F. Van Loan. *Matrix computations*. John Hopkins University Press, Baltimore, 1996.

[72] P. Grassberger. On the critical behavior of the general epidemic process and dynamical percolation. *Mathematical Biosciences*, 63:157–172, 1982.

[73] R. Guimerà and L. A. N. Amaral. Modeling the world-wide airport network. *European Journal of Physics B*, 38:381–385, 2003.

[74] R. Guimerà and L. A. N. Amaral. Cartography of complex networks: modules and universal roles. *Journal of Statistichal Mechanics: Theory and experiments*, P02001, 2005.

[75] R. Guimerà and L. A. N. Amaral. Functional cartography of complex metabolic networks. *Nature*, 433:895–900, 2005.

[76] R. Guimerà, M. Sales-Pardo, and L. A. N. Amaral. Modularity from fluctuations in random graphs and complex networks. *Phys. Rev. E*, 70:025101, 2004.

[77] T. Haliloglu, I. Bahar, and B. Erman. Gaussian dynamics of folded proteins. *Phys. Rev. Lett.*, 79:3090–3093, 1997.

[78] C. P. Herero. Self-avoiding walks on scale-free networks. *Phys. Rev. E*, 71:016103, 2005.

[79] P. Holme. Congestion and centrality in traffic flow on complex networks. *Advances in Complex Systems*, 6:163–176, 2003.

[80] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[81] P. Hu, G. Bader, D. A. wigle, and A. Emili. Computational prediction of cancer-gene function. *Nature*, 7:23–34, 2007.

[82] I. A. Hubner, J. Shimada, and E. I. Shakhnovich. Commitment and nucleation in the protein g transition state. *Jour. Mol. Biol.*, 336:745, 2003.

[83] Isaac A Hubner, Eric J Deeds, and Eugene I Shakhnovich. High-resolution protein folding with a transferable potential. *Proc Natl Acad Sci U S A*, 102(52):18914–9, 2005.

[84] I. T. Jolliffe. *Principle Component Analysis*. Springer, Mathematics, 2002.

[85] S. D. Kamvar, H. Haveliwala, T., C. D. Manning, and G. H. Golub. Exploiting the blockstructure of the web for computing pagerank. 117(23):10894–10903, 2003.

[86] B. J. Kim. Geographical coarse graining of complex networks. *Phys. Rev. Lett.*, 93(16):168701, 2004.

[87] K. Klemm and V.M. Erguiluz. Growing scale-free networks with small-world behavior. *Phys. Rev. E*, 65:057102, 2001.

[88] P.L. Krapivsky, S. Redner, and F. Leyvraz. Connectivity of growing random networks. *Phys. Rev. Lett.*, 85:4629–4632, 2000.

[89] P. Krishnaiah and L. Kanal. *Handbook of Statistics*. 1982.

[90] S. V. Krivov and M. Karplus. Free energy disconnectivity graphs: Application to peptide models. *J. Chem. Phys.*, 117(23):10894–10903, 2002.

[91] S. V. Krivov and M. Karplus. Hidden complexity of free energy surfaces for peptide (protein) folding. *Proc. Natl. Acad. Sci. USA*, 101(41):14766–14770, 2004.

[92] S. V. Krivov and M. Karplus. One-dimensional free-energy profiles of complex systems: Progress variables that preserve the barriers. *J Phys Chem B*, 110(25):12689–12698, 2006.

[93] S. V. Krivov and M. Karplus. private communications, 2007.

[94] O. Kurkcuoglua, R. L. Jerniganb, and P. Doruker. Mixed levels of coarse-graining of large proteins using elastic network model succeeds in extracting the slowest motions. *Polymer*, 45:649–657, 2003.

[95] S. Lafon and A. B. Lee. Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1393–1403, 2006.

[96] M. Latapy and P. Pons. Computing communities in large networks using random walks. *Proceedings Lecture Notes in Computer Science*, 3733:285–293, 2005.

[97] A. Lesne. Complex Networks: from Graph Theory to Biology. *Letters in Mathematical Physics*, 78:235–262, December 2006.

[98] M. Levitt and A. Warshel. Computer-simulation of protein folding. *Nature*, 253:694–698, 1975.

[99] A. Li and V. Daggett. Characterization of the transition state of protein unfolding by use of molecular dynamics: Chymotrypsin inhibitor 2. *PNAS*, 91:10430–10434, 1994.

[100] S. Lloyd. Least squares quantization in pcm. *IEEE Transaction Information Theory*, 28:129–138, 1982.

[101] A. Ma and A.R. Dinner. Automatic method for identifying reaction coordinates in complex systems. *J. Phys. Chem. B*, 109:6769–6779, 2005.

[102] B. B. Mandelbrot. *The fractal geometry of Nature*. Freeman, San Francisco, 1982.

[103] E. M. Marcotte, M. Pellegrini, M. J. Thompson, T. O. Yeates, and D. Eisenberg. A combined algorithm for genome-wide prediction of protein function. *Nature*, 402:83–86, 1999.

[104] A. Maritan, C. Micheletti, A. Trovato, and J. R. Banavar. Optimal shapes of compact strings. *Nature*, 406:287, 2000.

[105] S. J. Marrink, A. H. Vries, and A. E. Mark. Coarse grained model for semiquantitative lipid simulations. *J. Phys. Chem. B*, 108:750, 2004.

[106] S. Maslov and K. Sneppen. Specificity and stability in topology of protein networks. *Science*, 296:910–913, 2002.

[107] C. P. Massen and J. P. K. Doye. Characterizing the network topology of the energy landscapes of atomic clusters. *J. Chem. Phys.*, 122:084105, 2005.

[108] C. P. Massen and J. P. K. Doye. Identiftying communities within energy landscape. *PRE*, 71:046101, 2005.

[109] A. Matouschek, J.T. Kellis, L. Serrano, and A. R. Fersht. Mapping the transition state and pathway of protein folding by protein engineering. *Nature*, 340:122–126, 1989.

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

[111] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. *J. Chem. Phys.*, 21:1087, 1953.

[112] M. Molloy and B. Reed. A critical point for random graphs with a given degree sequence. *Random Structure Algorithm*, 6:161–179, 1995.

[113] M. Molloy and B. Reed. The size of the giant component of a random graph with a given degree sequence. *Comb. Probab. Comput.*, 7:295, 1998.

[114] D. Morton De La Chappelle. private communications, 2007.

[115] A. E. Motter, C. Zhou, and J. Kurths. Enhancing complex-network synchronization. *Europhysics Letters*, 69:334, 2005.

[116] Adilson E. Motter, Changsong Zhou, and Juergen Kurths. Network synchronization, diffusion, and the paradox of heterogeneity. *Phys. Rev. E*, 71:016116, 2005.

[117] S. Muff, F. Rao, and A. Caflisch. Local modularity measure for network clusterizations. *PRE*, 72:056107, 2005.

[118] V. Muñoz, Thompson P. A., J. Hofrichter, and W. A. Eaton. Folding dynamics and mechanism of beta-hairpin formation. *Nature*, 390:196–199, 1997.

[119] M. E. J. Newman. Scientific collaboration networks. ii. shortest paths, weighted networks, and centrality. *Phys. Rev. E*, 64:016132, 2001.

[120] M. E. J. Newman. Assortative mixing in networks. *Phys. Rev. Lett.*, 89:208701, 2002.

[121] M. E. J. Newman. The structure and functions of complex networks. *SIAM Rev.*, 45:167–256, 2003.

[122] M. E. J. Newman. Analysis of weighted networks. *Phys. Rev. E*, 70:056131, 2004.

[123] M. E. J. Newman. Detecting community structure in networks. *European Physical Journal B*, 38(2):321–330, 2004.

[124] M. E. J. Newman. Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, 69:066133, 2004.

[125] M. E. J. Newman. A measure of betweenness centrality based on random walks. *Social Networks*, 27:39–54, 2005.

[126] M. E. J. Newman. Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46(5):323–351, 2005.

[127] M. E. J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Phys. Rev. E*, 74(3):036104, 2006.

[128] M. E. J. Newman. Modularity and community structure in networks. *Proc. Natl. Acad. Sci. USA*, 103(23):8577–8582, 2006.

[129] M. E. J. Newman, Barabási A.-L., and J. D. Watts. *The Structure and Dynamics of Networks*. Princeton University Press, 2006.

[130] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Phys. Rev. E*, 69:026113, 2004.

[131] M. E. J. Newman and J. Park. Why social networks are different from other types of networks. *Phys. Rev. E*, 68:036122, 2003.

[132] T. Nishikawa, A. E. Motter, Y. Lai, and F. C. Hoppensteadt. Heterogeneity in oscillator networks: Are smaller worlds easier to synchronize? *Phys. Rev. Lett.*, 91(1):014101, 2003.

[133] J. D. Noh and H. Rieger. Random walks on complex networks. *Phys. Rev. Lett.*, 92(11):118701, 2004.

[134] E. Oh, K. Rho, H. Hong, and B. Kahng. Modular synchronization in complex networks. *Phys. Rev. E*, 72:047101, 2006.

[135] Anna Oliva, Adam Rosebrock, Francisco Ferrezuelo, Saumyadipta Pyne, Haiying Chen, Steve Skiena, Bruce Futcher, and Janet Leatherwood. The cell cycle–regulated genes of schizosaccharomyces pombe. *PLoS Biology*, 3, 2005.

[136] J.N. Onuchic, N.D. Socci, Z. Lutheyschulten, and P.G. Wolynes. Protein folding funnels: The nature of the transition state ensemble. *Folding & Design*, 1:441–450, 1996.

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

[138] V.S. Pande, A.Y. Grosberg, T. Tanaka, and D.S. Rokhsar. Pathways for protein folding: is a new view needed? *Current Opinion In Structural Biology*, 8:68–79, 1998.

[139] K. Park, Y-C. Lai, S. Gupte, and J-W. Kim. Synchronization in complex networks with a modular structure. *Chaos*, 16:015105, 2006.

[140] K. Park, Y-C. Lai, and N. Ye. Characterization of weighted complex networks. *Phys. Rev. E*, 70:026109, 2004.

[141] R. Pastor-Satorras and A. Vespignani. *Evolution and Structure of the internet: a statistical physics approach*. Cambridge university Press, 2004.

[142] X. Peng, L. D. Karuturi, R. K. M.and Miller, K. Lin, Y. H. Jia, P. Kondu, L. Wang, L. S. Wong, E. T. Liu, M. K. Balasubramanian, and J. H. Liu. Identification of cell cycle-regulated genes in fission yeast. *Journal of Mol. Biol.*, 16:1026–1042, 2005.

[143] T. Petermann and P. De Los Rios. Exploration of scale-free networks. *Eur. Phys. Jour. B*, 38:201, 2004.

[144] J-P. Pfister. Désambiguïsation d'un dictionnaire de synonymes. unpublished internship report, EPFL, 2000.

[145] F. Pontiggia, G. Colombo, C. Micheletti, and H. Orland. Anharmonicity and self-similarity of the free energy landscape of protein g. *Phys. Rev. Lett.*, 98:048102, 2007.

[146] D. J. S. Price. Networks of scientific papers. *Science*, 149:510–515, 1965.

[147] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *PNAS*, 101:2658–2663, 2004.

[148] F. Rao and A. Caflisch. The protein folding network. *J. Mol. Biol.*, 342:299–306, 2004.

[149] J. Reichardt and S. Bornholdt. Detecting fuzzy community structure in complex networks with a potts model. *Phys. Rev. Lett.*, 93:218701, 2004.

[150] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Phys. Rev. E.*, 74:016110, 2006.

[151] J. Reichardt and S. Bornholdt. When are networks truly modular. *Physica D-Nonlinear Phenomena*, 224:20–26, 2006.

[152] J. G. Restrepo, E. Ott, and B. R. Hunt. Onset of synchronization in large networks of coupled oscillators. *Phys. Rev. E*, 71:036151, 2005.

[153] J. G. Restrepo, E. Ott, and B. R. Hunt. Characterizing the dynamical importance of network nodes and links. *Phys. Rev. Lett.*, 97:094102, 2006.

[154] Y. M. Rhee and V. S. Pande. One-dimensional reaction coordinate and the corresponding potential of mean force from commitment probability. *J. Phys. Chem. B*, 109:6780, 2005.

[155] G. Rustici, J. Mata, K. Kivinen, P. Lió, C.J. Penkettand G. Burns, J. Hayles, A. Brazma, P. nurse, and J. Bähler. Periodic gene expression program of the fission yeast cell cycle. *Nature*, 36:809–817, 2004.

[156] M. Saerens, F. Fouss, L. Yen, and P. Dupont. The principal component analysis of a graph and its relationships to spectral clustering, 2004.

[157] G. Santoro. private communications, 2004.

[158] A. Scala, L. A. N. Amaral, and M. Barthélémy. Small-world networks and the conformation space of a short lattice polymer chain. *Europhysics Letters*, 55(4):594–600, 2001.

[159] M. Schaefer, C. Bartels, F. Leclerc, and M. Karplus. Effective atom volumes for implicit solvent models: comparison between Voronoi volumes and minimum fluctuation volumes. *Journal of Computational Chemistry*, 22(15):1857–1879, 2001.

[160] H. Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, 1998.

[161] N. Schwartz, R. Cohen, D. Ben-Avraham, and A-L. Barabsi. Percolation in directed scale-free networks. *Pys. Rev. E*, 66:015104, 2002.

[162] S. B. Seidman. Network structure and minimum degree. *Social Networks*, 5:269, 1983.

[163] H. A. Simon. *Biometrika*, 42:425, 1955.

[164] S. N. Soffer and A. Vazquez. Network clustering coefficient without degree-correlation biases. *Phys. Rev. E*, 71:057101, 2005.

[165] C. Song, S. Havlin, and H. A. Makse. Self-similarity of complex networks. *Nature*, 433:392–395, 2005.

[166] C. Song, S. Havlin, and H. A. Makse. Origins of fractality in the growth of complex networks. *Nature Physics*, 2:275–281, 2006.

[167] F.H. Stillinger. A topographic view of supercooled liquids and glass-formation. *Science*, 267:1935–1939, Mar 1995.

[168] S. H. Strogatz. Complex systems: Romanesque networks. *Nature*, 433:365–366, 2005.

[169] B. Tadic, S. Thurner, and G. J. Rodgers. Traffic on complex networks: Towards understanding global statistical properties from microscopic density fluctuations. *Phys. Rev. E*, 69:036102, 2004.

[170] J. R. Tyler and B. A. Wilkinson, D. M. Huberman. Dordrecht, 2003.

[171] Unknown. Book of Job, the Bible. Chapter 38-39.

[172] S. VanDongen. *Graph clustering by flow simulation.* PhD thesis, University of Utrecht, 2000. http://micans.org/mcl/.

[173] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Modeling of protein interaction networks. *Complexus*, 1:38–44, 2003.

[174] Vendruscolo, M. and Paci, E. Protein folding: Bringing theory and experiment closer together. *Curr. Opin. Struct. Biol.*, 13:82–87, 2003.

[175] Vendruscolo, M., Paci, E., Dobson, C. M., and Karplus, M. Three key residues form a critical contact network in a protein folding transition state. *Nature*, 409:641–645, 2001.

[176] T. Viscek. *Fractal Growth Phenomena*. World Scientific, Singapore, 1992.

[177] A. Wagner. The yeast protein interaction network evolves rapidly and contains few redundant duplicate genes. *Molecular Biology and Evolution*, 18:1283–1292, 2001.

[178] A. Warshel and M. Levitt. Folding and stability of helical proteins: carp myogen. *J. Mol. Biol.*, 106:421–437, 1976.

[179] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge University Press, 1994.

[180] D. J. Watts. *Small Worlds : The Dynamics of Networks between Order and Randomness*. Princeton University Press, 2003.

[181] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, 1998.

[182] D. M. Wilkinson and B. A. Huberman. A method for finding communities of related genes. *PNAS*, 101:5241–5248, 2004.

[183] J. C. Willis and G. U. Yule. Some statistics of evolution and geographical distribution in plants and animals, and their significance. *Nature*, 109:177–179, 1922.

[184] F. Wu and B. A. Huberman. Finding communities in linear time: a physics approach. *European Physical Journal B*, 38:331–338, 2004.

[185] F. Y. Wu. The potts model. *Review of Modern Physics*, 54:235–268, 1982.

[186] S. J. Yang. Exploring complex networks by walking on them. *Phys. Rev. E*, 71:016107, 2005.

[187] K. Y. Yeung and W. L. Ruzzo. Principal component analysis for clustering gene expression data. *Bioinformatics*, 17(9):763–774, 2001.

[188] S. H. Yook, H. Jeong, A-L. Barabási, and Y. Tu. Weighted evolving networks. *PRL*, 86(25):5835–5838, 2001.

[189] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33:452–473, 1977.

[190] M. Zamparo. private communications, 2007.

[191] S. H. Zhang, R. S. Wang, and X. S. Zhang. Identification of overlapping community structure in complex networks using fuzzy c-means clustering. *Physica A*, 374:483–490, 2007.

[192] F. Zhou, G. Grigoryan, S. R. Lustig, A. E. Keating, G. Ceder, and D. Morgan. Coarse-graining protein energetics in sequence variables. *Physical Review Letters*, 95:148103, 2005.

[193] H. Zhou. Distance, dissimilarity index, and network community structure. *Phys. Rev. E*, 67:061901, 2003.

[194] S. Zhou and R. J. Mondragón. Accurately modeling the internet topology. *Phys. Rev. E*, 70:066108, 2004.

Curriculum Vitae - David Gfeller

1. Coordinates

David Gfeller
Longemalle 26
1020 Renens
Switzerland
++41 21 625 68 48

Date of birth: 09 May 1980
Nationality: Swiss
Status: single
http://marie.epfl.ch/gfeller
david.gfeller@epfl.ch

2. Education

From 2004	PhD in Physics at the Laboratory of Statistical Biophysics at EPFL with Prof. Paolo De Los Rios.
2003	Master degree in Physics at University of Lausanne. Final mark: 5.8/6. 60 additional credits in Mathematics.
2001-2002	Exchange Student at Uppsala University (Sweden).
1998	High School Degree.

3. Teaching and Research Experience

Research:

From 2004	Research at EPFL about complex networks. Practical applications: linguistic networks, protein dynamics, stochastic processes on networks.
2003	Master Thesis: Stochastic Modelling of the protein import into mitochondria.
1998	2 months undergraduate research at University of Chicago in medical imaging at Dr. Hoffmann's Lab (presently at Buffalo University).

Teaching:

2004-2006	Supervising exercises in physics, theoretical biophysics and advanced statistical mechanics at EPFL
2003	Three months in Mali: Leading home-school of American childern.
2001-2002	Supervising physics lab for first year students at University of Lausanne.

4. Publications

1) D. Gfeller and P. De Los Rios, *Spectral coarse-graining of complex networks*, **Phys. Rev. Lett.** (2007) *in press.*

2) D. Gfeller, P. De Los Rios, D. Morton De La Chapelle, G. Caldarelli and F. Rao, *Uncovering the community structure of configuration space networks*, **Phys. Rev. E** (2007) *in press.*

3) D. Gfeller, P. De Los Rios, A. Caflisch and F. Rao, *Complex network analysis of free-energy landscape*, **Proc. Nat. Acad. Sci.**, 104, 1817-1822 (2007).

4) D. Gfeller, J.-C. Chappelier and P. De Los Rios, *Instabilities in the community structure of complex networks*, **Phys. Rev. E**, 72, 056135 (2005).

5) D. Gfeller, J.-C. Chappelier and P. De Los Rios, *Synonymy Dictionary improvement through Markov clustering and Clustering stability*, Proceeding of **ASMDA'05** (2005).

5. Oral talks, posters and internal report

1) *Complex network analysis of free-energy landscape*, oral talk at BIOWIRE, Cambridge, UK (2007).
2) *Inferring energy landscape from network topology*, oral talk at International School of Complexity, Biological Networks, Erice, Italy (2006).
3) *Inferring energy landscape from network community structure*, poster at the Workshop on Structure and Function of Biomolecule, Bedlewo, Poland (2006).
4) *Robustness of the community structure of networks*, oral talk at International Workshop on Structure and Function of Complex Networks, Torino, Italy (2006).
5) *Clustering instabilities*, poster at ECCS'05, Paris (2005).
6) *Analysis of a synonymy network*, internal report, EPFL (2004).

6. Academic honors

2003	University of Lausanne, Science Faculty Prize for excellent results in Master courses and Master Thesis.
1999	University of Lausanne, Science Faculty Prize for excellent results in first year of Physics.

7. Referee activity

1) Phys. Rev. E
2) European Journal of Physics B

8. Languages

1) French: mother tongue
2) English: Fluent
3) Italian: Fluent
4) Swedish: Fluent
5) German: Intermediate
6) Spanish: Intermediate

9. Computer skills

Word, Excel, PowerPoint, Latex, Matlab, Mathematica, C; Mac OS, Linux.