UNIVERSITAT DE BARCELONA

Mesoscopic descriptions of complex networks

Alberto Fernández Sabater

Mesoscopic Descriptions of Complex Networks

Autor
Alberto Fernández Sabater
Departament de Física Fonamental
Universitat de Barcelona

Juliol de 2008

Memòria presentada per optar al títol de
Doctor per la Univesitat de Barcelona

Programa de Doctorat de Física Avançada

Director
Sergio Gómez Jiménez
Professor Titular d'Universitat
Departament d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili

Codirector
Alex Arenas Moreno
Professor Titular d'Universitat
Departament d'Enginyeria Informàtica i Matemàtiques
Universitat Rovira i Virgili

Tutor
Albert Díaz Guilera
Professor Titular d'Universitat
Departament de Física Fonamental
Universitat de Barcelona

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In the last decades, scientists from several fields (including sociology, biology, physics, mathematics and computer science) have been building the new science of *complex networks*. From the Internet and the World Wide Web, to networks of friendships and even networks of disease transmissions, the reality of networks is almost everywhere in modern society. Scientists have found that many real systems have the structure of a complex network, i.e. a graph representative of the intricate connections that exist between its elements [11, 85, 87]. Then, the first question that arises is: what exactly is a network? The answer to this question is simple, because a network is nothing more than a set of elements (called nodes or vertices) and a set of links (also known as edges or arcs) that connect the elements of the network in pairs. Several common examples of complex networks can include: technological systems such as the Internet [1, 27] and the World Wide Web [2, 31]; biological systems such as gene or protein interaction networks [40, 46, 73]; a great variety of social networks [35, 36, 70]; financial markets [60]; and transport infrastructures such as railway and aerial routes [42] (see figure 1.1).

The new science of complex networks is important for various reasons. One of them is that, by focusing on the properties of real networks, it is concerned with the structure of networks as they arise naturally in the real world. Social networks and biological networks are naturally occurring networks of this kind, as are networks of information like citation networks and the World Wide Web. Furthermore, adequate theoretical models are also essential if the significance of any particular empirical finding is to be correctly understood. Hence, empirical observation and theoretic modeling continually stimulate each other.

Another distinctive feature of the science of complex networks is that it tries to establish the relationship between the structural properties of a networked system and its behavior. Complex networks not only have topological properties, but have dynamical properties as well. According to this view, the vertices of a network represent discrete dynamical entities, with their own rules of behavior, and the edges represent couplings between these entities.

The *macroscopic* description of complex networks in terms of statistical properties has been largely developed in the curse for a universal classification of them. Among these properties we find the *small world effect* describing that the average distance between nodes in a network is short, usually scaling logarithmically with the total number of nodes in the network. Another macroscopic

Figure 1.1: Example of a real complex network: the airports network, with data about passenger flights operating in the time period from 1 November 2000 to 31 October 2001, compiled by OAG Worldwide (Downers Grove, IL) and analyzed in [42].

property present in many complex networks is a characteristic power-law degree distribution, which means that there are typically many nodes with low degree and a small number with high degree, the precise distribution often following a power-law or exponential form. A third property that many networks have in common is network transitivity (or high clustering coefficient), which is the property that two nodes that are both neighbors of the same third node have a greater probability of also being neighbors of one another.

When complex networks are analyzed locally, some characteristics that become partially hidden in the statistical description emerge. The most relevant perhaps is the discovery of *community structure* in many of them [35], stating that nodes in a network are joined into groups of nodes connected between them strongly or densely, with sparse or weak connections between different groups. For instance, in the case of social networks (networks of friendships or other relations between individuals), it is commonly observed that such networks do have communities inside: subsets of nodes with quite dense node-node relations, whilst the relations between different subsets are not that dense.

The study of community structure in complex networks has deserved a lot of attention in recent years [22, 65]. In this work we exactly focus on the analysis of several *mesoscopic descriptions* of complex networks. It is an interesting subject matter because it can be very valuable identifying structures at a mesoscopic level of description which might reveal information about the functionality of groups of nodes [40, 43]: communities in social networks can represent real social groupings, maybe regarding common interests or studies; communities in citation networks can represent related articles on any common subject; communities in metabolic networks can represent cycles or other functional groupings; communities in the World Wide Web can represent pages about related subjects. The ability to identify these communities can help us to understand networks better and to analyze them more efficiently.

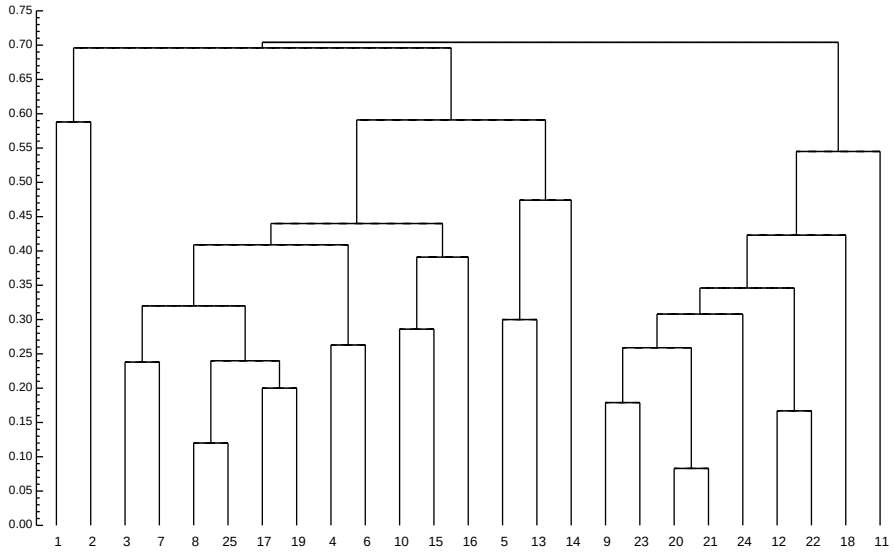


Figure 1.2: Example of a dendrogram depicting the hierarchical structure of a set with anorexia data and analyzed in [63].

Traditional methods for community detection

The traditional methods to detect community structure in networks are taken from the analysis of social networks and they are known with the generic name of *clustering* [20, 38, 84]. Clustering methods group individuals into groups of individuals or *clusters*, so that individuals in a cluster are close to one another. They are classified into two big groups, agglomerative and divisive, depending on whether they merge or split clusters respectively. Agglomerative methods have been more commonly used because they are more time effective than their divisive counterparts. In agglomerative hierarchical clustering, one starts defining a similarity measure between individuals and calculating the corresponding values between all pairs of individuals. Then, an iterative process begins from as many singleton clusters as individuals exist, merging the two more similar clusters at each repetition step until all individuals are in the same cluster. During this algorithmic process, a complete hierarchy of clusters is formed and it can be represented using a tree called *dendrogram* (see figure 1.2). Hierarchical clustering does not provide a single partition of the individuals into clusters, but the cuts through different levels in the tree provide clusters corresponding to a mesoscopic set of nested partitions.

Among the different types of agglomerative methods we find single linkage, complete linkage, unweighted average, weighted average, etc., which differ in the way they perform the iterative process that goes from the singleton clusters to the final one. Except for the single linkage case, all the other pair-group methods suffer from a problem of non-uniqueness when two or more similarity values between different clusters coincide during the amalgamation process. The traditional approach for solving this drawback has been to take any arbitrary

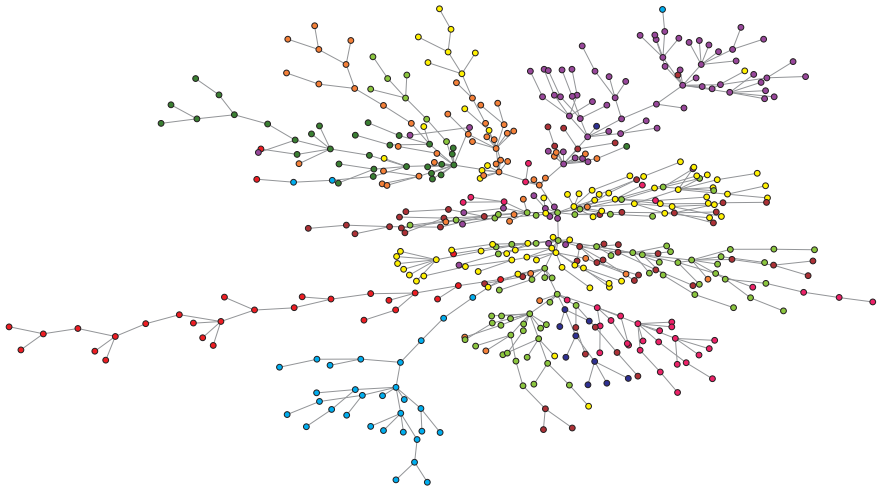


Figure 1.3: Minimum spanning tree for a portfolio of stocks from the S&P 500 index. Stocks are drawn with different colors corresponding to the ten industry sectors defined by the GICS (Global Industry Classification Standard).

criterion in order to break ties between similarity values, which results in different hierarchical classifications depending on the criterion followed. In chapter 2 we analyze several agglomerative hierarchical clustering methods, and we focus on the non-uniqueness problem in particular, proposing a new approach that solves the problem.

Financial complex networks

A particular type of complex networks are the correlation-based ones, i.e. networks used to visualize the structure of pair correlations among a set of variables. Specifically, starting from a set of variables one can calculate the correlation coefficient between all possible pairs. If we identify the different variables with the nodes of the network, each pair of nodes can be thought to be connected by an edge with a weight related to the correlation coefficient between the two variables. Such a network is therefore completely connected. Well known examples of correlation-based networks can be found in any portfolio of stocks traded in a financial market, by considering the evolution of the time series obtained from the daily difference of the closure stock price.

Sometimes there is a need to filter such complex networks into simpler relevant subnetworks. In [60], Mantegna detected a hierarchical structure present in a portfolio of stocks traded in a financial market. The goal of the study was to obtain the taxonomy of a portfolio of stocks by using the information of time series of stock prices only. Starting from the correlation coefficient matrix of a set of stocks, one can obtain a metric distance and then identify the clusters of companies by means of the *minimum spanning tree* (see figure 1.3), which is equivalent to the single linkage hierarchical tree in order to obtain the *subdominant ultrametric* [77].

Another important application of the hierarchical clustering techniques is certainly in *portfolio selection*. Many attempts have been made to solve this central problem, from the classical approach of Markowitz [61]. In chapter 3, after an initial description of financial complex networks, we focus on the portfolio selection problem. We consider a generalization of the standard Markowitz mean-variance model which includes cardinality and bounding constraints. These constraints ensure the investment in a given number of different assets and limit the amount of capital to be invested in each asset. By considering this model, the portfolio selection problem becomes a mixed quadratic and integer programming problem and, hence, there is no exact method able to solve the problem in an efficient way. In a first proposal, we show how asset trees, in addition to their ability to form economically meaningful clusters, can contribute to the portfolio selection problem. Additionally, we give a new heuristic method based on artificial neural networks in order to trace out the efficient frontier associated to the portfolio selection problem.

Quality functions for evaluation of modular structure

There are two main approaches to perform the grouping step of any clustering procedure. One can use hierarchical algorithms and obtain a nested series of partitions, or use partitional algorithms which produce a single partition of the data. In spite of their differences, both techniques share a common point in the use of a *quality function*, i.e. a quantitative criterion to evaluate how good partitions are. In hierarchical clustering such a function is needed to know which of the partitions in the hierarchy is the best one, and partitional clustering usually produces clusters by optimizing a quality function.

In terms of computation, the community detection problem is very similar to that of finding a ground state of a spin glass model. A spin glass is a disordered material exhibiting high magnetic frustration on account of the inability of the system to remain in a single lowest energy state (the ground state). Reichardt and Bornholdt [79] set the basis for a unified framework under which community detection may be viewed. They showed that the problem can be mapped onto finding the ground state of an infinite range Potts spin glass, where the similarity measures are translated into coupling strengths, the energy of the system is interpreted as the quality function of the partition into communities, and the spin states are the community indices.

In chapter 4 we describe one of the most successful quality functions for community detection, the *modularity*, under a unified framework of quality functions coming out from a particular spin glass model. Modularity was proposed by Newman and Girvan [69] as a function to compute the quality of each partition along a dendrogram, and to search for local optima pointing at satisfactory partitions. However, provided that the optimization of modularity is a NP-hard problem, it cannot be performed by exhaustive search and only optimization heuristics have proved to be competent in finding suboptimal solutions of the modularity function in feasible computational time. Here we propose an exact procedure for size reduction of complex networks preserving the value of modularity. The use of this size reduction allows to search in a more exhaustive

way through the partitions space, what usually will end in improved values of modularity compared to those obtained without using the size reduction.

Descriptions at different scales

Returning to the example of the portfolio with stocks from the S&P 500 index, in figure 1.3 we had the stocks classified in 10 groups according to the different sectors defined by the GICS (Global Industry Classification Standard): energy, materials, industrials, consumer discretionary, consumer staples, health care, financials, information technology, telecommunication services, and utilities. However, this classification is just one among a great variety of possibilities. As a matter of fact, the same GICS gives three additional levels of classification for the industry. Thus, one can classify a portfolio of stocks in: 10 sectors, 24 industry groups, 59 industries, or 112 sub-industries.

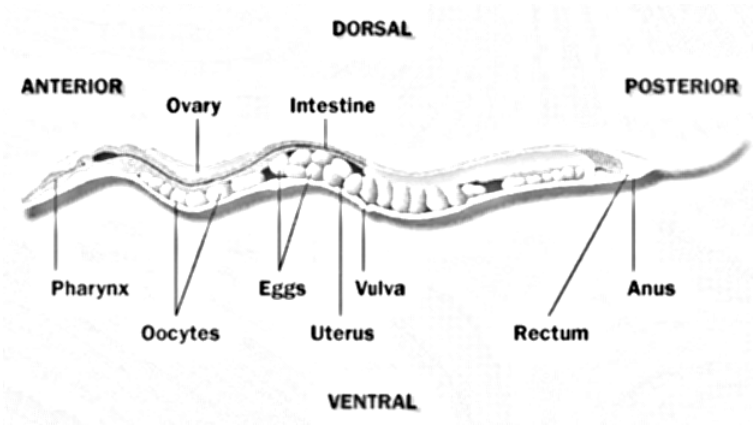
The existence of several scales of description is not just a peculiarity of financial systems, but a common feature in many real complex systems. In chapter 5, motivated by the recent finding that the optimization of modularity has a resolution limit related to the characteristic scale imposed by the total strength (sum of weights) of the network, we introduce a multiple resolution method that allows the process of optimizing modularity to find community structure at different scales of description. We apply the method to unravel the mesoscales of the neuronal connectivity of the *nematode C. elegans*. The whole nervous system of this worm can be represented as an adjacency network (see figure 1.4). The purpose of the analysis is to find any correlation between the substructures prevailing in the mesoscales and the functionalities in the worm.

General descriptions for groups of nodes

The analysis of modular structure using the modularity quality function provides a partition of the network into communities, where each community is a subset of nodes more connected between them than with the rest of the nodes in the network. However, modularity is strongly focused on communities, and therefore it cannot be used to detect general groups of nodes revealed by alternative connectivity patterns. Although a lot of work has been done to devise reliable techniques to optimize modularity, very little has been done to analyze the concept of modularity itself and its reliability as a method for detection of more general modular structure.

In chapter 6 we propose a general framework to describe groups of nodes in networks using *motifs* (small connected subnetworks) as elementary units. In particular, we give several definitions for groups of nodes, including communities, based on the principle that they contain more motifs than a null model representing a randomized version of the network at study. Thus, we develop the mathematical formulation for extensions of modularity where the building blocks are different types of motifs (e.g. triangles, cycles and paths between nodes), and not just edges as in the original expression of modularity.

(a)



(b)

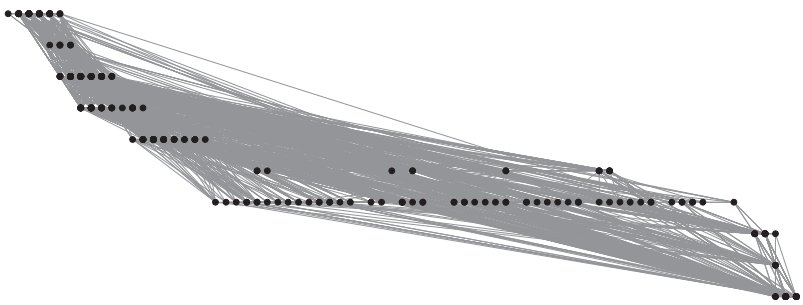


Figure 1.4: (a) *C. elegans* anatomy; and (b) network with its neuronal connectivity, where neurons are horizontally arranged according to their spatial position and vertically arranged according to the ten different ganglia in the worm.

Chapter 2

Hierarchical Clustering

Clustering is the organization of a collection of individuals into groups of individuals or *clusters* based on a measure of similarity, and so that individuals within a cluster are more similar to each other than they are to any individual belonging to a different cluster. Typical clustering procedures involve the following two steps [48]: first, definition of a proximity measure appropriate to the data domain; and second, grouping.

Proximity between individuals can be measured by a similarity function, or alternatively it can be measured by a distance function used to reflect dissimilarity between individuals. The grouping step can be performed following two different schemes: *hierarchical* and *partitional*. Hierarchical clustering algorithms produce a nested series of partitions based on a criterion for merging or splitting clusters based on similarity. Partitional clustering algorithms obtain a single partition of the data instead of a clustering structure. Partitional techniques usually produce clusters by optimizing a quality function, although combinatorial search for an optimum value of the quality function is computationally prohibitive. In practice, therefore, heuristic algorithms are typically run multiple times with different starting points, and the best configuration obtained from all of the runs is used as the output clustering.

This classification can be supplemented by several issues that may affect the different approaches [49].

- Agglomerative *vs.* divisive: This aspect relates to algorithmic structure and operation. An agglomerative approach begins with each individual in a distinct (singleton) cluster, and successively merges clusters together until a stopping criterion is satisfied. A divisive method begins with all the individuals in a single cluster and keeps splitting clusters until a stopping criterion is met.

- Exclusive *vs.* nonexclusive: An exclusive classification is a partition of the set of individuals, i.e. each object belongs to exactly one cluster. A nonexclusive (or overlapping) classification can assign each individual to several clusters simultaneously.

- Deterministic *vs.* stochastic: This issue is most relevant to partitional approaches designed to optimize a quality function. This optimization can

be accomplished using random decisions, in which case we are performing a stochastic search. Otherwise, the clustering method is said to be deterministic.

In this chapter we focus on agglomerative hierarchical clustering techniques [20, 38, 84], which are deterministic (in other chapters some stochastic techniques are developed). Only exclusive approaches are considered here and, therefore, we always deal with non-overlapping clusters. Agglomerative hierarchical clustering starts from a proximity matrix between individuals, each one forming a singleton cluster. Then, clusters are themselves grouped into groups of clusters or *superclusters*, the process being repeated until a complete hierarchy is formed. In section 2.1 we describe the classic pair-group agglomerative algorithm.

However, agglomerative hierarchical clustering methods suffer from a problem of non-uniqueness when the pair-group algorithm is used and two or more distances between different clusters coincide during the amalgamation process. The traditional approach for solving this drawback has been to take any arbitrary criterion in order to break ties between distances, which results in different hierarchical classifications depending on the criterion followed. In section 2.2 we propose a variable-group algorithm that consists in grouping more than two clusters at the same time when ties occur [29]. In the same section we introduce a tree representation for the results of the algorithm, which we call a *multidendrogram*, and in section 2.3 we show some results corresponding to data from a real example.

Among the different types of agglomerative methods we find single linkage, complete linkage, unweighted average, weighted average, etc., which differ in the definition of the proximity measure between clusters. In section 2.4 we describe several hierarchical clustering strategies and we give their generalization according to the variable-group approach. Section 2.5 goes one step further and explains the reason of the differences in the results obtained using one agglomerative hierarchical method or another. Finally, in section 2.6 we generalize Lance and Williams' formula, which enables the implementation of the hierarchical clustering algorithm in a recursive way.

2.1 Pair-group agglomerative algorithm

Agglomerative hierarchical procedures build a hierarchical classification in a bottom-up way, from a proximity matrix containing dissimilarity data between individuals of a set $\Omega = \{x_1, \ldots, x_n\}$ (the same analysis could be done using similarity data). The algorithm has the following steps:

0) Initialize n singleton clusters with one individual in each of them: $\{x_1\}$, $\ldots$, $\{x_n\}$. Initialize also the distances between clusters, $D(\{x_i\}, \{x_j\})$, with the values of the distances between individuals, $d(x_i, x_j)$:

$$D(\{x_i\}, \{x_j\}) = d(x_i, x_j)\,, \qquad \forall i, j = 1, \ldots, n\,.$$

1) Find the shortest distance separating two different clusters.

2) Select two clusters X_i and $X_{i'}$ separated by such shortest distance and merge them into a new supercluster $X_i \cup X_{i'}$.

3) Compute the distances $D(X_i \cup X_{i'}, X_j)$ between the new supercluster $X_i \cup X_{i'}$ and each of the other clusters X_j.

4) If all individuals are not in a single cluster yet, then go back to step 1.

Following Sneath and Sokal [84], this type of approach is known as a *pair-group* method, in opposition to *variable-group* methods which will be discussed in section 2.2. Depending on the criterion used for the calculation of distances in step 3, we can implement different agglomerative hierarchical methods. In this chapter we study some of the most commonly used ones, which are: single linkage, complete linkage, unweighted average, weighted average, unweighted centroid, weighted centroid and joint between-within.

The use of any hierarchical clustering technique on a finite set Ω with n individuals results in an *n-tree* on Ω, which is defined as a subset T of parts of Ω satisfying the following conditions:

(i) $\Omega \in T$,

(ii) $\emptyset \notin T$,

(iii) $\forall x \in \Omega \quad \{x\} \in T$,

(iv) $\forall X, Y \in T \quad (X \cap Y = \emptyset \quad \vee \quad X \subseteq Y \quad \vee \quad Y \subseteq X)$.

An n-tree gives only the hierarchical structure of a classification, but the use of a hierarchical clustering technique also associates a height h with each of the clusters obtained. All this information is gathered in the definition of a *valued tree* on Ω, which is a pair (T, h) where T is an n-tree on Ω and $h : T \longrightarrow \mathbb{R}$ is a function such that $\forall X, Y \in T$:

(i) $h(X) \geq 0$,

(ii) $h(X) = 0 \quad \Longleftrightarrow \quad |X| = 1$,

(iii) $X \subsetneq Y \quad \Longrightarrow \quad h(X) < h(Y)$,

where $|X|$ denotes the cardinality of X.

2.2 Variable-group agglomerative algorithm

2.2.1 Non-uniqueness problem

The problem of non-uniqueness may arise at step 2 of the pair-group algorithm in section 2.1, when two or more pairs of clusters are separated by the shortest distance value (i.e., the shortest distance is tied). Every choice for breaking ties may have important consequences, because it changes the collection of clusters and the distances between them, possibly resulting in different hierarchical classifications. It must be noted here that not all tied distances will produce ambiguity: they have to be the shortest ones and they also have to involve a common cluster. On the other hand, ambiguity is not limited to cases with ties in the original proximity values, but ties may arise during the clustering process too.

For example, suppose that we have a graph with four individuals like that of figure 2.1, where the initial distance between any two individuals is the value

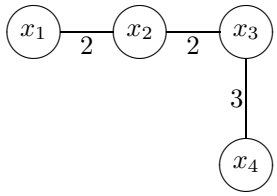


Figure 2.1: Toy graph with four individuals and shortest path distances.

of the shortest path connecting them. This means, for instance, that the initial distance between x_2 and x_4 is equal to 5. Using the unweighted average criterion, we can obtain three different valued trees. The graphical representation of valued trees are the so called *dendrograms*, and figure 2.2 shows the three corresponding dendrograms obtained for our toy example. The first two dendrograms are quite similar, but the third one shows a considerably different hierarchical structure. Hence, if the third dendrogram is the only one obtained by a software package, one could extract from it the wrong conclusion that x_3 is closer to x_4 than it is to x_2.

Except for the single linkage case, all the other clustering techniques suffer from a non-uniqueness problem, sometimes called the *ties in proximity* problem, which is caused by ties either occurring in the initial proximity data or arising during the amalgamation process. From the family of agglomerative hierarchical methods, complete linkage is more susceptible than other methods to encounter ties during the clustering process, since it does not originate new proximity values different from the initial ones. With regard to the presence of ties in the original data, they are more frequent when one works with binary variables, or even with integer variables comprising just some few distinct values. However, they can also appear using continuous variables, specially if the precision of experimental data is low. Sometimes, on the contrary, the absence of ties might be due to the representation of data with more decimal digits than it should be done. The non-uniqueness problem also depends on the measure used to obtain the proximity values from the initial variables. Moreover, in general, the larger the data set, the more ties arise [59].

The ties in proximity problem is well-known from several studies in different fields, for example in biology [9, 10, 45], in psychology [90], or in chemistry [59].

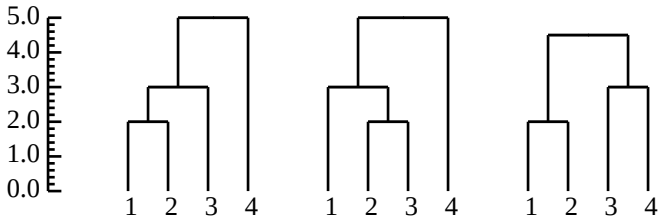


Figure 2.2: Unweighted average dendrograms for the toy example.

Nevertheless, this problem is frequently ignored in software packages [10, 63, 90], and those packages which do not ignore it fail to adopt a common standard with respect to ties. Many of them simply break the ties in any arbitrary way, thus producing a single hierarchy. In some cases the analysis is repeated a given number of times with randomized input data order, and then additional criteria can be used for selecting one of the possible solutions [9]. In other cases, some requirements are given on the number of individuals and the number of characteristics needed to generate proximity data without ties [45, 59]. None of these proposals can ensure the complete absence of ties, neither can all their requirements be satisfied always.

Another possibility for dealing with multiple solutions is to use further criteria, like a distortion measure [20], and select the best solution among all the possible ones. The result of this approach will depend necessarily on the distortion measure used, which means that an additional choice must be made. However, this proposal does not ensure the uniqueness of the solution, since several candidate solutions might share the same minimum distortion value. Besides, in ill conditioned problems (those susceptible to the occurrence of too many ties), it is not feasible to perform an exhaustive search for all possible hierarchical classifications, due to its high computational cost. With regard to this, in [90] two data sets are analyzed using many random permutations of the input data order, and with additional criteria the quality of each solution is evaluated. It is observed that the best solutions frequently emerge after many permutations, and it is also noticed that the goodness of these solutions necessarily depends on the number of permutations used.

An alternative proposal is to seek a hierarchical classification which describes common structure among all the possible solutions, as recommended in [45]. One approach is to prune as little as possible from the classifications being compared to arrive at a common structure such as the maximal common pruned tree [63]. Care must be taken not to prune too much, so this approach can be followed only when the number of alternative solutions is small and they are all known. Furthermore, the maximal common pruned tree need not be uniquely defined and it does not give a complete classification for all the individuals under study.

2.2.2 Variable-group approach: the multidendrogram

Any decision taken to break ties in the toy graph of figure 2.1 would be arbitrary. In fact, the use of an unfortunate rule might lead us to the worst dendrogram of the three. A logical solution to the pair-group criterion problem might be to assign the same importance to all tied distances and, therefore, to use a variable-group criterion. In our example of figure 2.1 this means the amalgamation of individuals x_1, x_2 and x_3 in a single cluster at the same time. The immediate consequence is that we have to calculate the distance between the new cluster $\{x_1\} \cup \{x_2\} \cup \{x_3\}$ and the cluster $\{x_4\}$. In the unweighted average case this distance is equal to 5, that is, the arithmetic mean among the values 7, 5 and 3, corresponding respectively to the distances $D(\{x_1\}, \{x_4\})$, $D(\{x_2\}, \{x_4\})$ and $D(\{x_3\}, \{x_4\})$. We must also decide what height should be assigned to the new cluster formed by x_1, x_2 and x_3, which could be any value between the minimum and the maximum distances that separate any two of them. In this case the minimum distance is 2 and corresponds to both of the tied distances $D(\{x_1\}, \{x_2\})$ and $D(\{x_2\}, \{x_3\})$, while the maximum distance

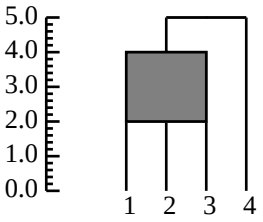


Figure 2.3: Unweighted average multidendrogram for the toy example.

is the one separating x_1 from x_3 and it is equal to 4.

Following the variable-group criterion on a finite set Ω with n individuals, we no longer get several valued trees, but we obtain a unique tree which we call a *multivalued tree* on Ω, and we define it as a triplet (T, h_l, h_u) where T is an n-tree on Ω and $h_l, h_u : T \longrightarrow \mathbb{R}$ are two functions such that $\forall X, Y \in T$:

(i) $0 \leq h_l(X) \leq h_u(X)$,

(ii) $h_l(X) = 0 \iff h_u(X) = 0 \iff |X| = 1$,

(iii) $X \subsetneq Y \implies h_l(X) < h_l(Y)$.

A multivalued tree associates with every cluster X in the hierarchical classification two height values, $h_l(X)$ and $h_u(X)$, corresponding respectively to the lower and upper bounds at which member individuals can be merged into cluster X. When $h_l(X)$ and $h_u(X)$ coincide for every cluster X, the multivalued tree is just a valued tree. On the contrary, when there is any cluster X for which $h_l(X) < h_u(X)$, it is like having multiple valued trees because every selection of a height $h(X)$ inside the interval $[h_l(X), h_u(X)]$ corresponds to a different valued tree. The length of the interval indicates the degree of heterogeneity inside cluster X. We also introduce here the concept of *multidendrogram* to refer to the graphical representation of a multivalued tree. In figure 2.3 we show the corresponding multidendrogram for the toy example. The shadowed region between heights 2 and 4 refers to the interval between the respective values of h_l and h_u for cluster $\{x_1\} \cup \{x_2\} \cup \{x_3\}$, which in turn also correspond to the minimum and maximum distances separating any two of the constituent clusters $\{x_1\}$, $\{x_2\}$ and $\{x_3\}$.

Let us consider the situation shown in figure 2.4, where nine different clusters are to be grouped into superclusters. The clusters to be amalgamated should be those separated by the shortest distance. The picture shows the edges connecting clusters separated by such shortest distance, so we observe that there are six pairs of clusters separated by shortest edges. A pair-group clustering algorithm typically would select any of these pairs, for instance (X_8, X_9), and then it would compute the distance between the new supercluster $X_8 \cup X_9$ and the rest of the clusters X_i, for all $i \in \{1, 2, \ldots, 7\}$. What we propose here is to follow a variable-group criterion and create as many superclusters as groups of clusters connected by shortest edges. In figure 2.4, for instance, the nine initial clusters would be grouped into the four following superclusters: X_1, $X_2 \cup X_3$, $X_4 \cup X_5 \cup X_6$ and $X_7 \cup X_8 \cup X_9$. Then, all the pairwise distances between

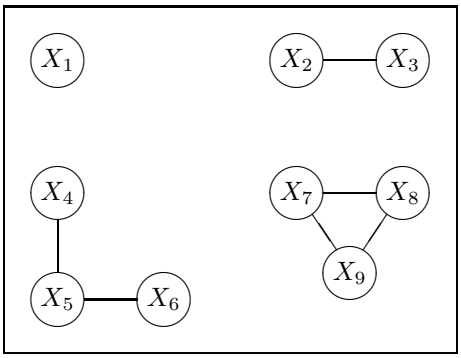


Figure 2.4: Simultaneous occurrence of different superclusters.

the four superclusters should be computed. In general, we must be able to compute distances $D(X_I, X_J)$ between any two superclusters $X_I = \bigcup_{i\in I} X_i$ and $X_J = \bigcup_{j\in J} X_j$, each one of them made up of several clusters indexed by $I = \{i_1, i_2, \ldots, i_p\}$ and $J = \{j_1, j_2, \ldots, j_q\}$, respectively.

The algorithm that we propose in order to ensure uniqueness in agglomerative hierarchical clustering has the following steps:

0) Initialize n singleton clusters with one individual in each of them: $\{x_1\}$, $\ldots$, $\{x_n\}$. Initialize also the distances between clusters, $D(\{x_i\}, \{x_j\})$, with the values of the distances between individuals, $d(x_i, x_j)$:

$$D(\{x_i\}, \{x_j\}) = d(x_i, x_j)\,, \qquad \forall i, j = 1, \ldots, n\,.$$

1) Find the shortest distance separating two different clusters, and record it as D_{lower}.

2) Select all the groups of clusters separated by shortest distance D_{lower} and merge them into several new superclusters X_I. The result of this step can be some superclusters made up of just one single cluster ($|I| = 1$), as well as some superclusters made up of various clusters ($|I| > 1$). Notice that the latter superclusters all must satisfy the condition $D_{min}(X_I) = D_{lower}$, where

$$D_{min}(X_I) = \min_{i\in I} \min_{\substack{i'\in I \\ i'\neq i}} D(X_i, X_{i'})\,.$$

3) Update the distances between clusters following the next substeps:

   3.1) Compute the distances $D(X_I, X_J)$ between all superclusters, and record the minimum of them as D_{next} (this will be the shortest distance D_{lower} in the next iteration of the algorithm).

   3.2) For each supercluster X_I made up of various clusters ($|I| > 1$), assign a common amalgamation interval $[D_{lower}, D_{upper}]$ for all its constituent clusters X_i, $i \in I$, where $D_{upper} = D_{max}(X_I)$ and

$$D_{max}(X_I) = \max_{i\in I} \max_{\substack{i'\in I \\ i'\neq i}} D(X_i, X_{i'})\,.$$

4) If all individuals are not in a single cluster yet, then go back to step 1.

Using the pair-group algorithm, only the centroid methods (weighted and unweighted) may produce *reversals*. Let us remember that a reversal arises in a valued tree when it contains at least two clusters X and Y for which $X \subset Y$ but $h(X) > h(Y)$ [63]. In the case of the variable-group algorithm, reversals may appear in substep 3.2. Although reversals make dendrograms difficult to interpret if they occur during the last stages of the agglomeration process, it can be argued that they are not very disturbing if they occur during the first stages. Thus, as happens with the centroid methods in the pair-group case, it could be reasonable to use the variable-group algorithm as long as no reversals at all or only unimportant ones were produced.

Sometimes, in substep 3.2 of the variable-group clustering algorithm, it will not be enough to adopt a fusion interval, but it will be necessary to obtain an exact fusion value (e.g., in order to calculate a distortion measure). In these cases, given the lower and upper bounds at which the tied clusters can merge into a supercluster, one possibility is to select the fusion value naturally suggested by the method being applied. For instance, in the case of the toy example and the corresponding multidendrogram shown in figures 2.1 and 2.3, the fusion value would be 2.7 (the unweighted average distance). If the clustering method used was a different one such as single linkage or complete linkage, then the fusion value would be 2 or 4, respectively. Another possibility is to use systematically the shortest distance as the fusion value for the tied clusters. Both criteria allow the recovering of the pair-group result for the single linkage method. The latter criterion, in addition, avoids the appearance of reversals. However, it must be emphasized that the adoption of exact fusion values, without considering the fusion intervals at their whole lengths, means that some valuable information regarding the heterogeneity of the clusters is being lost.

2.3 Soils example

We show here a real example which has been studied by Morgan and Ray [63] using the complete linkage method. It is the *Glamorganshire soils* example, formed by similarity data between 23 different soils. A table with the similarities can be found also in Morgan and Ray [63], where the values are given with an accuracy of three decimal places. In order to work with dissimilarities, first of all we have transformed the similarities $s(x_i, x_j)$ into the corresponding dissimilarities $d(x_i, x_j) = 1 - s(x_i, x_j)$.

The original data present a tied value for pairs of soils (3,15) and (3,20), which is responsible for two different dendrograms using the complete linkage strategy. We show them in figure 2.5. Morgan and Ray [63] explain that the 23 soils have been categorized into eight "great soil groups" by a surveyor. Focusing on soils 1, 2, 6, 12 and 13, which are the only members of the brown earths soil group, we see that the dendrogram in figure 2.5a does not place them in the same cluster until they join soils from five other soil groups, forming the cluster (1, 2, 3, 20, 12, 13, 15, 5, 6, 8, 14, 18). From this point of view, the dendrogram in figure 2.5b is better, since the corresponding cluster loses soils 8, 14 and 18, each representing a different soil group. So, in this case, we have two possible solution dendrograms and the probability of obtaining the "good" one is, hence, 50%.

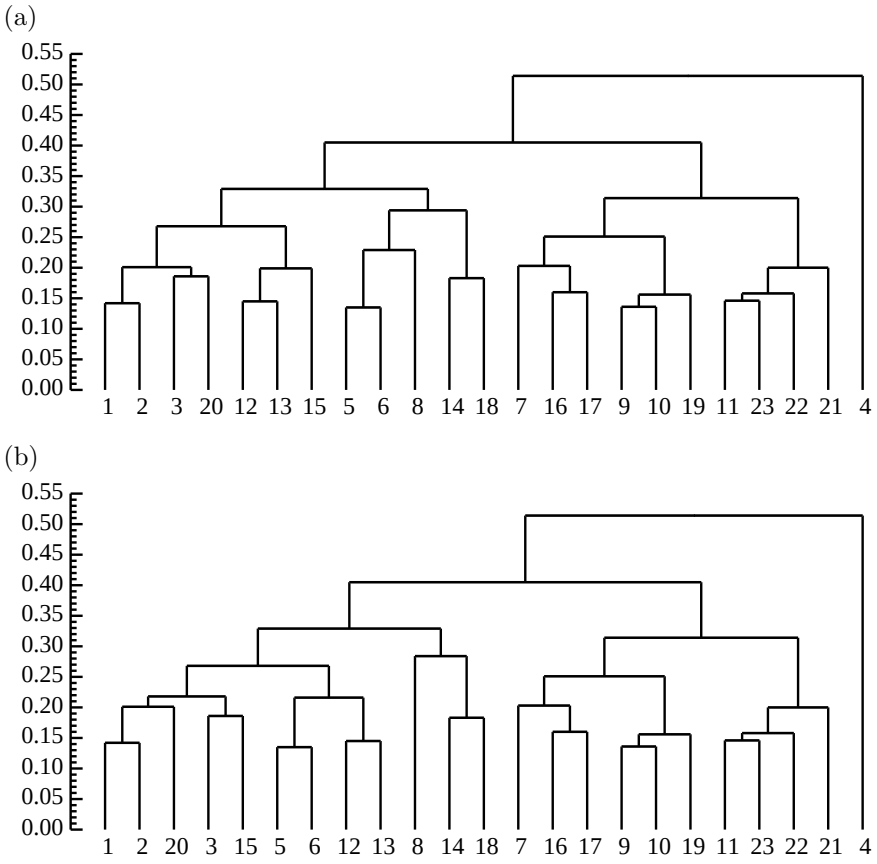


Figure 2.5: Complete linkage dendrograms for the soils data. According to the brown earths soil group formed by soils 1, 2, 6, 12 and 13, the dendrogram in (a) is worse than that in (b) because the former joins these soils at a posterior stage of the amalgamation process.

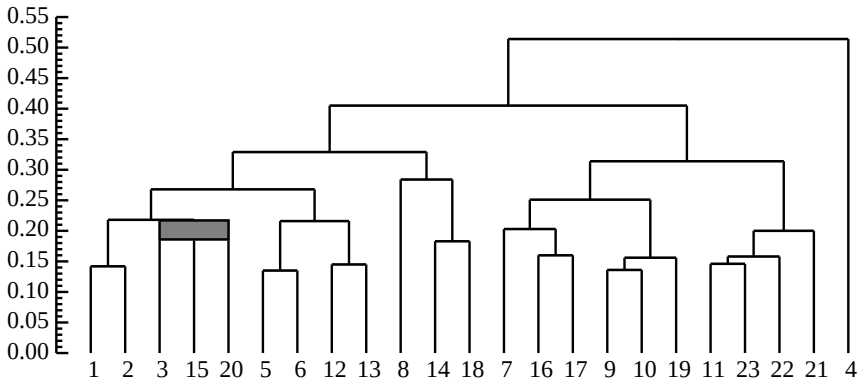


Figure 2.6: Complete linkage multidendrogram for the soils data (with an accuracy of three decimal places).

On the other hand, in figure 2.6 we can see the multidendrogram corresponding to the *Glamorganshire soils* data. The existence of a tie comprising soils 3, 15 and 20 is clear from this tree representation. Besides, the multidendrogram gives us the good classification, that is, the one with soils 8, 14 and 18 out of the brown earths soil group. Except for the internal structure of the cluster (1, 2, 3, 15, 20), the rest of the multidendrogram hierarchy coincides with that of the dendrogram shown in figure 2.5(b).

Finally, notice that the incidence of ties depends on the accuracy with which proximity values are available. In this example, if dissimilarities had been measured to four decimal places, then the tie causing the non-unique complete linkage dendrogram might have disappeared. On the contrary, the probability of ties is higher if lower accuracy data are used. For instance, when we consider the same soils data but with an accuracy of only two decimal places, we obtain the multidendrogram shown in figure 2.7, where three different ties can be observed.

2.4 Agglomerative hierarchical methods

In the variable-group clustering algorithm previously proposed we have seen the necessity of agglomerating simultaneously two families of clusters, respectively indexed by $I = \{i_1, i_2, \ldots, i_p\}$ and $J = \{j_1, j_2, \ldots, j_q\}$, into two superclusters $X_I = \bigcup_{i \in I} X_i$ and $X_J = \bigcup_{j \in J} X_j$. In the following subsections we derive, for each of the most commonly used agglomerative hierarchical clustering strategies, the distance between the two superclusters, $D(X_I, X_J)$, in terms of the distances between the respective component clusters, $D(X_i, X_j)$.

2.4.1 Single linkage

In *single linkage* clustering, also called *nearest neighbor* or *minimum* method, the distance between two clusters X_i and X_j is defined as the distance between

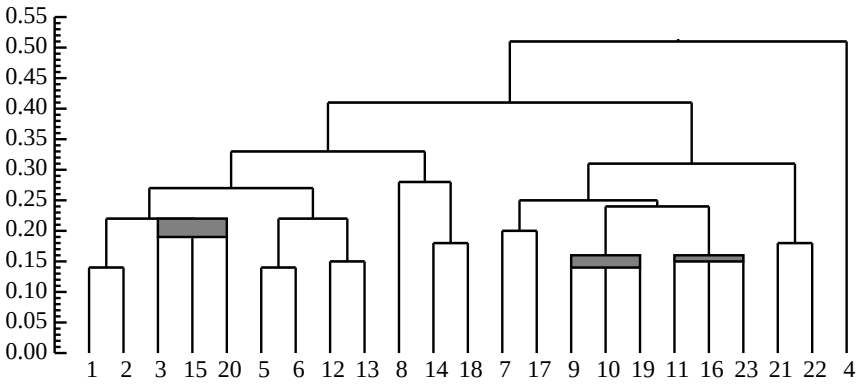


Figure 2.7: Complete linkage multidendrogram for the soils data, with an accuracy of two decimal places.

the closest pair of individuals, one in each cluster:

$$D(X_i, X_j) = \min_{x \in X_i} \min_{y \in X_j} d(x, y)\,. \tag{2.1}$$

This means that the distance between two superclusters X_I and X_J can be defined as

$$D(X_I, X_J) = \min_{x \in X_I} \min_{y \in X_J} d(x, y) = \min_{i \in I} \min_{x \in X_i} \min_{j \in J} \min_{y \in X_j} d(x, y)\,. \tag{2.2}$$

Notice that this formulation generalizes the definition of distance between clusters in the sense that equation (2.1) is recovered from equation (2.2) when $|I| = |J| = 1$, that is, when superclusters I and J are both composed of a single cluster. Grouping terms and using the definition in equation (2.1), we get the equivalent definition:

$$D(X_I, X_J) = \min_{i \in I} \min_{j \in J} D(X_i, X_j)\,. \tag{2.3}$$

2.4.2 Complete linkage

In *complete linkage* clustering, also known as *furthest neighbor* or *maximum* method, cluster distance is defined as the distance between the most remote pair of individuals, one in each cluster:

$$D(X_i, X_j) = \max_{x \in X_i} \max_{y \in X_j} d(x, y)\,. \tag{2.4}$$

Starting from equation (2.4) and following the same reasoning as in the single linkage case, we extend the definition of distance to the superclusters case as

$$D(X_I, X_J) = \max_{i \in I} \max_{j \in J} D(X_i, X_j)\,. \tag{2.5}$$

2.4.3 Unweighted average

Unweighted average clustering, also known as *group average* method or *UPGMA* (Unweighted Pair-Group Method using Averages), iteratively forms clusters made up of pairs of previously formed clusters, based on the arithmetic mean distances between their member individuals. It uses an unweighted averaging procedure, that is, when clusters are joined to form a larger cluster, the distance between this new cluster and any other cluster is calculated weighting each individual in those clusters equally, regardless of the structural subdivision of the clusters:

$$D(X_i, X_j) = \frac{1}{|X_i||X_j|} \sum_{x \in X_i} \sum_{y \in X_j} d(x, y)\,. \tag{2.6}$$

When the variable-group strategy is followed, the UPGMA name of the method should be modified to that of *UVGMA* (Unweighted Variable-Group Method using Averages), and the distance definition between superclusters in this case should be

$$\begin{aligned} D(X_I, X_J) &= \frac{1}{|X_I||X_J|} \sum_{x \in X_I} \sum_{y \in X_J} d(x, y) \\ &= \frac{1}{|X_I||X_J|} \sum_{i \in I} \sum_{x \in X_i} \sum_{j \in J} \sum_{y \in X_j} d(x, y)\,. \end{aligned} \tag{2.7}$$

Using equation (2.6), we get the desired definition in terms of the distances between component clusters:

$$D(X_I, X_J) = \frac{1}{|X_I||X_J|} \sum_{i \in I} \sum_{j \in J} |X_i||X_j| D(X_i, X_j)\,. \tag{2.8}$$

In this case, $|X_I|$ is the number of individuals in supercluster X_I, that is, $|X_I| = \sum_{i \in I} |X_i|$.

2.4.4 Weighted average

In *weighted average* strategy, also called *WVGMA* (Weighted Variable-Group Method using Averages) in substitution of the corresponding pair-group name *WPGMA*, we calculate the distance between two superclusters X_I and X_J by taking the arithmetic mean of the pairwise distances, not between individuals in the original matrix of distances, but between component clusters in the matrix used in the previous iteration of the procedure:

$$D(X_I, X_J) = \frac{1}{|I||J|} \sum_{i \in I} \sum_{j \in J} D(X_i, X_j)\,. \tag{2.9}$$

This method is related to the unweighted average one in that the former derives from the latter when we consider

$$|X_i| = 1 \quad \forall i \in I \qquad \text{and} \qquad |X_j| = 1 \quad \forall j \in J\,. \tag{2.10}$$

It weights the most recently admitted individuals in a cluster equally to its previous members. The weighting discussed here is with reference to individuals composing a cluster and not to the average distances in Lance and Williams' recursive formula (see section 2.6), in which equal weights apply for weighted clustering and different weights apply for unweighted clustering [84].

2.4.5 Unweighted centroid

The next three clustering techniques assume that individuals can be represented by points in Euclidean space. This method and the next one further assume that the measure of dissimilarity between any pair of individuals is the squared Euclidean distance between the corresponding pair of points. When the dissimilarity between two clusters X_i and X_j is defined to be the squared distance between their centroids, we are performing *unweighted centroid* (or simply *centroid*) clustering, also called *UPGMC* (Unweighted Pair-Group Method using Centroids):

$$D(X_i, X_j) = \|\overline{x}_i - \overline{x}_j\|^2 , \tag{2.11}$$

where $\overline{x}_i$ and $\overline{x}_j$ are the centroids of the points in clusters X_i and X_j respectively, and $\|\cdot\|$ is the Euclidean norm. Therefore, under the variable-group point of view, the method could be named *UVGMC* and the distance between two superclusters can be generalized to the definition:

$$D(X_I, X_J) = \|\overline{x}_I - \overline{x}_J\|^2 . \tag{2.12}$$

Next we prove that this definition can be expressed in terms of equation (2.11) as

$$\begin{aligned} D(X_I, X_J) = {} & \frac{1}{|X_I||X_J|} \sum_{i\in I} \sum_{j\in J} |X_i||X_j| D(X_i, X_j) \\ & - \frac{1}{|X_I|^2} \sum_{i\in I} \sum_{\substack{i'\in I \\ i'>i}} |X_i||X_{i'}| D(X_i, X_{i'}) \\ & - \frac{1}{|X_J|^2} \sum_{j\in J} \sum_{\substack{j'\in J \\ j'>j}} |X_j||X_{j'}| D(X_j, X_{j'}) . \end{aligned} \tag{2.13}$$

Certainly, given a cluster X_i, its centroid is

$$\overline{x}_i = \frac{1}{|X_i|} \sum_{x\in X_i} x ,$$

and the centroid of a supercluster X_I can be expressed in terms of its constituent centroids by the equation:

$$\overline{x}_I = \frac{1}{|X_I|} \sum_{i\in I} |X_i| \overline{x}_i . \tag{2.14}$$

Now, given two superclusters X_I and X_J, the distance between them defined in equation (2.12) is

$$D(X_I, X_J) = \|\overline{x}_I - \overline{x}_J\|^2 = \|\overline{x}_I\|^2 + \|\overline{x}_J\|^2 - 2\langle \overline{x}_I, \overline{x}_J \rangle ,$$

where $\langle \cdot, \cdot \rangle$ stands for the inner product. If we substitute each centroid by its

definition (2.14), we obtain:

$$\begin{aligned} D(X_I, X_J) = &\frac{1}{|X_I|^2} \sum_{i\in I} \sum_{i'\in I} |X_i||X_{i'}| \langle \overline{x}_i, \overline{x}_{i'} \rangle \\ &+ \frac{1}{|X_J|^2} \sum_{j\in J} \sum_{j'\in J} |X_j||X_{j'}| \langle \overline{x}_j, \overline{x}_{j'} \rangle \\ &- \frac{1}{|X_I||X_J|} \sum_{i\in I} \sum_{j\in J} |X_i||X_j| 2\langle \overline{x}_i, \overline{x}_j \rangle \,. \end{aligned}$$

Now, since

$$2\langle \overline{x}_i, \overline{x}_j \rangle = \|\overline{x}_i\|^2 + \|\overline{x}_j\|^2 - \|\overline{x}_i - \overline{x}_j\|^2 \,,$$

we have that

$$\begin{aligned} D(X_I, X_J) = &\frac{1}{|X_I|^2} \sum_{i\in I} \sum_{i'\in I} |X_i||X_{i'}| \langle \overline{x}_i, \overline{x}_{i'} \rangle \\ &+ \frac{1}{|X_J|^2} \sum_{j\in J} \sum_{j'\in J} |X_j||X_{j'}| \langle \overline{x}_j, \overline{x}_{j'} \rangle \\ &- \frac{1}{|X_I||X_J|} \sum_{i\in I} \sum_{j\in J} |X_i||X_j| \|\overline{x}_i\|^2 - \frac{1}{|X_I||X_J|} \sum_{i\in I} \sum_{j\in J} |X_i||X_j| \|\overline{x}_j\|^2 \\ &+ \frac{1}{|X_I||X_J|} \sum_{i\in I} \sum_{j\in J} |X_i||X_j| \|\overline{x}_i - \overline{x}_j\|^2 \,. \end{aligned}$$

This can be rewritten as

$$\begin{aligned} D(X_I, X_J) = &\frac{1}{|X_I||X_J|} \sum_{i\in I} \sum_{j\in J} |X_i||X_j| \|\overline{x}_i - \overline{x}_j\|^2 \\ &- \frac{1}{|X_I|} \sum_{i\in I} |X_i| \|\overline{x}_i\|^2 + \frac{1}{|X_I|^2} \sum_{i\in I} \sum_{i'\in I} |X_i||X_{i'}| \langle \overline{x}_i, \overline{x}_{i'} \rangle \\ &- \frac{1}{|X_J|} \sum_{j\in J} |X_j| \|\overline{x}_j\|^2 + \frac{1}{|X_J|^2} \sum_{j\in J} \sum_{j'\in J} |X_j||X_{j'}| \langle \overline{x}_j, \overline{x}_{j'} \rangle \,, \end{aligned}$$

and, grouping terms,

$$\begin{aligned} D(X_I, X_J) = &\frac{1}{|X_I||X_J|} \sum_{i\in I} \sum_{j\in J} |X_i||X_j| \|\overline{x}_i - \overline{x}_j\|^2 \\ &- \frac{1}{|X_I|^2} \sum_{i\in I} \sum_{i'\in I} |X_i||X_{i'}| \left(\|\overline{x}_i\|^2 - \langle \overline{x}_i, \overline{x}_{i'} \rangle \right) \\ &- \frac{1}{|X_J|^2} \sum_{j\in J} \sum_{j'\in J} |X_j||X_{j'}| \left(\|\overline{x}_j\|^2 - \langle \overline{x}_j, \overline{x}_{j'} \rangle \right) \,. \end{aligned}$$

The second and third terms can be simplified a little more, thanks to the equality

$$\begin{aligned} &\sum_{i\in I} \sum_{i'\in I} |X_i||X_{i'}| \left(\|\overline{x}_i\|^2 - \langle \overline{x}_i, \overline{x}_{i'} \rangle \right) = \\ &\quad = \sum_{i\in I} \sum_{\substack{i'\in I \\ i'>i}} |X_i||X_{i'}| \left(\|\overline{x}_i\|^2 + \|\overline{x}_{i'}\|^2 - 2\langle \overline{x}_i, \overline{x}_{i'} \rangle \right) \,. \end{aligned}$$

With this simplification, we have that

$$D(X_I, X_J) = \frac{1}{|X_I||X_J|} \sum_{i \in I} \sum_{j \in J} |X_i||X_j| \|\overline{x}_i - \overline{x}_j\|^2$$
$$- \frac{1}{|X_I|^2} \sum_{i \in I} \sum_{\substack{i' \in I \\ i' > i}} |X_i||X_{i'}| \|\overline{x}_i - \overline{x}_{i'}\|^2$$
$$- \frac{1}{|X_J|^2} \sum_{j \in J} \sum_{\substack{j' \in J \\ j' > j}} |X_j||X_{j'}| \|\overline{x}_j - \overline{x}_{j'}\|^2 ,$$

and, recalling the definition of distance between two clusters given in equation (2.11), we finally obtain the desired form of equation (2.13).

2.4.6 Weighted centroid

In *weighted centroid* strategy, also called *median* method or *WVGMC* (Weighted Variable-Group Method using Centroids) in substitution of the pair-group name *WPGMC*, we modify the definition of dissimilarity between two clusters given in the unweighted centroid case, assigning each cluster the same weight in calculating the "centroid". Now the center of a supercluster X_I is the average of the centers of the constituent clusters:

$$\overline{x}_I = \frac{1}{|I|} \sum_{i \in I} \overline{x}_i \,. \tag{2.15}$$

This method is related to the unweighted centroid one by relation (2.10), which also related the weighted average strategy to the corresponding unweighted average. So, in this case we define the distance between two superclusters as

$$D(X_I, X_J) = \frac{1}{|I||J|} \sum_{i \in I} \sum_{j \in J} D(X_i, X_j)$$
$$- \frac{1}{|I|^2} \sum_{i \in I} \sum_{\substack{i' \in I \\ i' > i}} D(X_i, X_{i'}) - \frac{1}{|J|^2} \sum_{j \in J} \sum_{\substack{j' \in J \\ j' > j}} D(X_j, X_{j'}) \,. \tag{2.16}$$

2.4.7 Joint between-within

Székely and Rizzo [88] propose an agglomerative hierarchical clustering method that minimizes a *joint between-within* cluster distance, measuring both heterogeneity between clusters and homogeneity within clusters. This method extends Ward's minimum variance method [92] by defining the distance between two clusters X_i and X_j in terms of any power $\alpha \in (0, 2]$ of Euclidean distances between individuals:

$$D(X_i, X_j) = \frac{|X_i||X_j|}{|X_i| + |X_j|} \left(\frac{2}{|X_i||X_j|} \sum_{x \in X_i} \sum_{y \in X_j} \|x - y\|^\alpha \right.$$
$$\left. - \frac{1}{|X_i|^2} \sum_{x \in X_i} \sum_{x' \in X_i} \|x - x'\|^\alpha - \frac{1}{|X_j|^2} \sum_{y \in X_j} \sum_{y' \in X_j} \|y - y'\|^\alpha \right) . \tag{2.17}$$

When $\alpha = 2$, cluster distances are a weighted squared distance between cluster centers

$$D(X_i, X_j) = \frac{2|X_i||X_j|}{|X_i| + |X_j|} \|\overline{x}_i - \overline{x}_j\|^2 , \tag{2.18}$$

equal to twice the cluster distance that is used in Ward's method.

Next we derive the following recursive formula for updating cluster distances as a generalization of equation (2.17):

$$\begin{aligned} D(X_I, X_J) = {} & \frac{1}{|X_I| + |X_J|} \sum_{i \in I} \sum_{j \in J} (|X_i| + |X_j|) D(X_i, X_j) \\ & - \frac{|X_J|}{|X_I|(|X_I| + |X_J|)} \sum_{i \in I} \sum_{\substack{i' \in I \\ i' > i}} (|X_i| + |X_{i'}|) D(X_i, X_{i'}) \\ & - \frac{|X_I|}{|X_J|(|X_I| + |X_J|)} \sum_{j \in J} \sum_{\substack{j' \in J \\ j' > j}} (|X_j| + |X_{j'}|) D(X_j, X_{j'}) . \end{aligned} \tag{2.19}$$

We give here a proof based on that of Székely and Rizzo [88] for their agglomerative hierarchical formulation. Using the following constants:

$$\begin{aligned} \theta_{ij} &= \frac{1}{|X_i||X_j|} \sum_{x \in X_i} \sum_{y \in X_j} \|x - y\|^\alpha , \\ \theta_{ii} &= \frac{1}{|X_i|^2} \sum_{x \in X_i} \sum_{x' \in X_i} \|x - x'\|^\alpha , \end{aligned} \tag{2.20}$$

the definition (2.17) of distance between two clusters X_i and X_j is

$$D(X_i, X_j) = \frac{|X_i||X_j|}{|X_i| + |X_j|} (2\theta_{ij} - \theta_{ii} - \theta_{jj}) .$$

Consider now the superclusters X_I and X_J formed by merging clusters X_i, for all $i \in I$, and X_j, for all $j \in J$. Define the corresponding constants:

$$\begin{aligned} \theta_{IJ} &= \frac{1}{|X_I||X_J|} \sum_{x \in X_I} \sum_{y \in X_J} \|x - y\|^\alpha \\ &= \frac{1}{|X_I||X_J|} \sum_{i \in I} \sum_{j \in J} \sum_{x \in X_i} \sum_{y \in X_j} \|x - y\|^\alpha , \\ \theta_{II} &= \frac{1}{|X_I|^2} \sum_{x \in X_I} \sum_{x' \in X_I} \|x - x'\|^\alpha \\ &= \frac{1}{|X_I|^2} \sum_{i \in I} \sum_{i' \in I} \sum_{x \in X_i} \sum_{x' \in X_{i'}} \|x - x'\|^\alpha \\ &= \frac{1}{|X_I|^2} \sum_{i \in I} \left(\sum_{x \in X_i} \sum_{x' \in X_i} \|x - x'\|^\alpha + 2 \sum_{\substack{i' \in I \\ i' > i}} \sum_{x \in X_i} \sum_{x' \in X_{i'}} \|x - x'\|^\alpha \right) , \end{aligned}$$

so that in terms of the original constants (2.20) we have:

$$\begin{aligned}\theta_{IJ} &= \frac{1}{|X_I||X_J|}\sum_{i\in I}\sum_{j\in J}|X_i||X_j|\theta_{ij}\,,\\ \theta_{II} &= \frac{1}{|X_I|^2}\sum_{i\in I}\left(|X_i|^2\theta_{ii}+2\sum_{\substack{i'\in I\\ i'>i}}|X_i||X_{i'}|\theta_{ii'}\right).\end{aligned}$$

Therefore, the distance between superclusters X_I and X_J is given by

$$\begin{aligned}D(X_I,X_J) &= \frac{|X_I||X_J|}{|X_I|+|X_J|}(2\theta_{IJ}-\theta_{II}-\theta_{JJ})\\ &= \frac{|X_I||X_J|}{|X_I|+|X_J|}\left[\frac{2}{|X_I||X_J|}\sum_{i\in I}\sum_{j\in J}|X_i||X_j|\theta_{ij}\right.\\ &\quad -\frac{1}{|X_I|^2}\sum_{i\in I}\left(|X_i|^2\theta_{ii}+2\sum_{\substack{i'\in I\\ i'>i}}|X_i||X_{i'}|\theta_{ii'}\right)\\ &\quad \left.-\frac{1}{|X_J|^2}\sum_{j\in J}\left(|X_j|^2\theta_{jj}+2\sum_{\substack{j'\in J\\ j'>j}}|X_j||X_{j'}|\theta_{jj'}\right)\right].\end{aligned}$$

Simplify

$$\begin{aligned}&\sum_{i\in I}\left(|X_i|^2\theta_{ii}+2\sum_{\substack{i'\in I\\ i'>i}}|X_i||X_{i'}|\theta_{ii'}\right)=\\ &= \sum_{i\in I}\left[|X_i|^2\theta_{ii}+\sum_{\substack{i'\in I\\ i'>i}}|X_i||X_{i'}|(2\theta_{ii'}-\theta_{ii}-\theta_{i'i'}+\theta_{ii}+\theta_{i'i'})\right]\\ &= \sum_{i\in I}\left[|X_i|^2\theta_{ii}+\sum_{\substack{i'\in I\\ i'>i}}|X_i||X_{i'}|(\theta_{ii}+\theta_{i'i'})+\sum_{\substack{i'\in I\\ i'>i}}(|X_i|+|X_{i'}|)D(X_i,X_{i'})\right]\\ &= |X_I|\sum_{i\in I}|X_i|\theta_{ii}+\sum_{i\in I}\sum_{\substack{i'\in I\\ i'>i}}(|X_i|+|X_{i'}|)D(X_i,X_{i'})\,,\end{aligned}$$

where in last equality we have used the equivalence

$$\sum_{i\in I}\left[|X_i|^2\theta_{ii}+\sum_{\substack{i'\in I\\ i'>i}}|X_i||X_{i'}|(\theta_{ii}+\theta_{i'i'})\right]=|X_I|\sum_{i\in I}|X_i|\theta_{ii}\,.$$

Hence,

$$
\begin{aligned}
(|X_I| + |X_J|)D(X_I, X_J) = 2 \sum_{i \in I} \sum_{j \in J} |X_i||X_j|\theta_{ij} \\
-|X_J| \sum_{i \in I} |X_i|\theta_{ii} - \frac{|X_J|}{|X_I|} \sum_{i \in I} \sum_{\substack{i' \in I \\ i' > i}} (|X_i| + |X_{i'}|)D(X_i, X_{i'}) \\
-|X_I| \sum_{j \in J} |X_j|\theta_{jj} - \frac{|X_I|}{|X_J|} \sum_{j \in J} \sum_{\substack{j' \in J \\ j' > j}} (|X_j| + |X_{j'}|)D(X_j, X_{j'}) \,,
\end{aligned}
$$

or, equivalently,

$$
\begin{aligned}
(|X_I| + |X_J|)D(X_I, X_J) = \sum_{i \in I} \sum_{j \in J} |X_i||X_j|2\theta_{ij} \\
- \sum_{i \in I} |X_i|\theta_{ii} \sum_{j \in J} |X_j| - \sum_{i \in I} |X_i| \sum_{j \in J} |X_j|\theta_{jj} \\
-\frac{|X_J|}{|X_I|} \sum_{i \in I} \sum_{\substack{i' \in I \\ i' > i}} (|X_i| + |X_{i'}|)D(X_i, X_{i'}) \\
-\frac{|X_I|}{|X_J|} \sum_{j \in J} \sum_{\substack{j' \in J \\ j' > j}} (|X_j| + |X_{j'}|)D(X_j, X_{j'}) \,,
\end{aligned}
$$

which is also the same as

$$
\begin{aligned}
(|X_I| + |X_J|)D(X_I, X_J) = \sum_{i \in I} \sum_{j \in J} (|X_i| + |X_j|)D(X_i, X_j) \\
-\frac{|X_J|}{|X_I|} \sum_{i \in I} \sum_{\substack{i' \in I \\ i' > i}} (|X_i| + |X_{i'}|)D(X_i, X_{i'}) \\
-\frac{|X_I|}{|X_J|} \sum_{j \in J} \sum_{\substack{j' \in J \\ j' > j}} (|X_j| + |X_{j'}|)D(X_j, X_{j'}) \,.
\end{aligned}
$$

And this is exactly the desired formulation given in equation (2.19).

2.5 Space distortion

Markedly different results can be obtained when a set of objects is clustered using distinct agglomerative hierarchical methods. The point is that clustering criteria are not model-free, but implicitly specify models for data and can provide misleading summaries of the class structure present in the data. Dubien and Warde [24] formalized the idea of *space distortion* in hierarchical clustering, referring to strategies as *space-conserving*, *space-contracting* or *space-dilating*. Suppose that X_i and $X_{i'}$ are two clusters selected during the pair-group algorithm of section 2.1 to be merged into the supercluster $X_i \cup X_{i'}$, and let X_j

represent each one of the other clusters whose distance to the new supercluster $X_i \cup X_{i'}$ has to be calculated. Then, a clustering strategy is said to be space-conserving if

$$\begin{aligned} \min\{D(X_i, X_j), D(X_{i'}, X_j)\} &< D(X_i \cup X_{i'}, X_j) \\ \max\{D(X_i, X_j), D(X_{i'}, X_j)\} &> D(X_i \cup X_{i'}, X_j)\,, \end{aligned} \tag{2.21}$$

and to be space-contracting if the first inequality is broken, and space-dilating if the second inequality is broken.

Next, we modify properly equation (2.21) in terms of the variable-group algorithm of section 2.2. Let X_I and X_J be two superclusters whose distance of separation has to be computed. Then, we can define a clustering strategy to be space-conserving if

$$\min_{i\in I}\min_{j\in J} D(X_i, X_j) < D(X_I, X_J) < \max_{i\in I}\max_{j\in J} D(X_i, X_j)\,, \tag{2.22}$$

and to be space-contracting if the first inequality is broken, and space-dilating if the second inequality is broken.

The complete linkage clustering method is space-dilating and, as a consequence, it generally leads to tight clusters that join others only with difficulty and at relatively high dissimilarity values. From their definition, complete linkage classes are compact but need not be externally isolated. On the contrary, the single linkage clustering method is space-contracting, leading frequently to long untidy clusters. From their definition, single linkage classes are isolated from one another but need not possess much internal cohesion. The elongate growth of single linkage clusters is known as *chaining effect* [48]: different individuals merge into a large cluster almost one at a time during the iterative amalgamation process. Furthermore, the chaining effect not only arises in the agglomerative hierarchical methods studied in this chapter, but it is also observed in other clustering methods. For example, in [21] the authors showed that Newman's Fast algorithm for community detection [66] tends to favor the creation of large communities (clusters) at the expense of smaller ones, specially at the early stages of the grouping process.

The single linkage and complete linkage clustering criteria take extreme approaches to the aim that classes be externally isolated and internally cohesive, each of them concentrating on satisfying one of these objectives respectively. The two average linkage clustering methods, weighted and unweighted, take a middle road between these extremes and are space-conserving. Weighted clustering was only introduced in an attempt to give merging branches in a (multi)dendrogram equal weight regardless of the number of individuals carried on each branch. Such a procedure weights the individuals unequally. Unweighted clustering gives equal weight to each individual in clusters whose distance from another cluster is being evaluated.

On the other hand, clustering strategies that can lead to reversals, such as the two centroid linkage methods, sometimes are said to be *highly space-contracting* since one can have, in terms of the pair-group algorithmic approach,

$$D(X_i \cup X_{i'}, X_j) < D(X_i, X_{i'}) < \min\{D(X_i, X_j), D(X_{i'}, X_j)\}\,, \tag{2.23}$$

and in variable-group algorithmic terms,

$$D(X_I, X_J) < D_{lower} < \min_{i\in I}\min_{j\in J} D(X_i, X_j)\,, \tag{2.24}$$

where the reader needs to remember that D_{lower} was the shortest distance separating two different clusters inside X_I or X_J. The joint between-within clustering method can be said to be space-dilating because it can lead to

$$D(X_i \cup X_{i'}, X_j) > \max\{D(X_i, X_j), D(X_{i'}, X_j)\}, \tag{2.25}$$

in terms of the pair-group algorithm, or

$$D(X_I, X_J) > \max_{i \in I} \max_{j \in J} D(X_i, X_j), \tag{2.26}$$

using the variable-group notation.

2.6 Recursive formulation

Lance and Williams [55] put the most commonly used agglomerative hierarchical strategies into a single system, avoiding the necessity of a separate computer program for each of them. Assume three clusters X_i, $X_{i'}$ and X_j, containing $|X_i|$, $|X_{i'}|$ and $|X_j|$ individuals respectively and with distances between them already determined as $D(X_i, X_{i'})$, $D(X_i, X_j)$ and $D(X_{i'}, X_j)$. Further assume that the smallest of all distances still to be considered is $D(X_i, X_{i'})$, so that X_i and $X_{i'}$ are joined to form a new supercluster $X_i \cup X_{i'}$, with $|X_i| + |X_{i'}|$ individuals. Lance and Williams express $D(X_i \cup X_{i'}, X_j)$ in terms of the distances already defined, all known at the moment of fusion, using the following recurrence relation:

$$\begin{aligned} D(X_i \cup X_{i'}, X_j) = {} & \alpha_i D(X_i, X_j) + \alpha_{i'} D(X_{i'}, X_j) \\ & + \beta D(X_i, X_{i'}) + \gamma |D(X_i, X_j) - D(X_{i'}, X_j)| . \end{aligned} \tag{2.27}$$

With this technique superclusters can always be computed from previous clusters and it is not necessary to return to the original dissimilarity data during the clustering process. The values of the parameters α_i, $\alpha_{i'}$, β and γ determine the nature of the sorting strategy. Table 2.1 gives the values of the parameters that define the most commonly used agglomerative hierarchical clustering methods.

We next give a generalization of formula (2.27) compatible with the amalgamation of more than two clusters simultaneously. Suppose that one wants to agglomerate two superclusters X_I and X_J, respectively indexed by $I = \{i_1, i_2, \ldots, i_p\}$ and $J = \{j_1, j_2, \ldots, j_q\}$. We define the distance between them as:

$$\begin{aligned} D(X_I, X_J) = {} & \sum_{i \in I} \sum_{j \in J} \alpha_{ij} D(X_i, X_j) \\ & + \sum_{i \in I} \sum_{\substack{i' \in I \\ i' > i}} \beta_{ii'} D(X_i, X_{i'}) + \sum_{j \in J} \sum_{\substack{j' \in J \\ j' > j}} \beta_{jj'} D(X_j, X_{j'}) \\ & + \delta \sum_{i \in I} \sum_{j \in J} \gamma_{ij} [D_{max}(X_I, X_J) - D(X_i, X_j)] \\ & - (1 - \delta) \sum_{i \in I} \sum_{j \in J} \gamma_{ij} [D(X_i, X_j) - D_{min}(X_I, X_J)] , \end{aligned} \tag{2.28}$$

where

$$D_{max}(X_I, X_J) = \max_{i \in I} \max_{j \in J} D(X_i, X_j)$$

Table 2.1: Parameter values for the Lance and Williams' formula.

Method	α_i ($\alpha_{i'}$)	β	γ
Single linkage	$\frac{1}{2}$	0	$-\frac{1}{2}$
Complete linkage	$\frac{1}{2}$	0	$+\frac{1}{2}$
Unweighted average	$\frac{\lvert X_i\rvert}{\lvert X_i\rvert+\lvert X_{i'}\rvert}$	0	0
Weighted average	$\frac{1}{2}$	0	0
Unweighted centroid	$\frac{\lvert X_i\rvert}{\lvert X_i\rvert+\lvert X_{i'}\rvert}$	$-\frac{\lvert X_i\rvert\lvert X_{i'}\rvert}{(\lvert X_i\rvert+\lvert X_{i'}\rvert)^2}$	0
Weighted centroid	$\frac{1}{2}$	$-\frac{1}{4}$	0
Joint between-within	$\frac{\lvert X_i\rvert+\lvert X_j\rvert}{\lvert X_i\rvert+\lvert X_{i'}\rvert+\lvert X_j\rvert}$	$-\frac{\lvert X_j\rvert}{\lvert X_i\rvert+\lvert X_{i'}\rvert+\lvert X_j\rvert}$	0

Table 2.2: Parameter values for the variable-group formula.

Method	α_{ij}	$\beta_{ii'}$ ($\beta_{jj'}$)	γ_{ij}	δ
Single linkage	$\frac{1}{\lvert I\rvert\lvert J\rvert}$	0	$\frac{1}{\lvert I\rvert\lvert J\rvert}$	0
Complete linkage	$\frac{1}{\lvert I\rvert\lvert J\rvert}$	0	$\frac{1}{\lvert I\rvert\lvert J\rvert}$	1
Unweighted average	$\frac{\lvert X_i\rvert\lvert X_j\rvert}{\lvert X_I\rvert\lvert X_J\rvert}$	0	0	–
Weighted average	$\frac{1}{\lvert I\rvert\lvert J\rvert}$	0	0	–
Unweighted centroid	$\frac{\lvert X_i\rvert\lvert X_j\rvert}{\lvert X_I\rvert\lvert X_J\rvert}$	$-\frac{\lvert X_i\rvert\lvert X_{i'}\rvert}{\lvert X_I\rvert^2}$	0	–
Weighted centroid	$\frac{1}{\lvert I\rvert\lvert J\rvert}$	$-\frac{1}{\lvert I\rvert^2}$	0	–
Joint between-within	$\frac{\lvert X_i\rvert+\lvert X_j\rvert}{\lvert X_I\rvert+\lvert X_J\rvert}$	$-\frac{\lvert X_J\rvert}{\lvert X_I\rvert}\frac{\lvert X_i\rvert+\lvert X_{i'}\rvert}{\lvert X_I\rvert+\lvert X_J\rvert}$	0	–

and

$$D_{min}(X_I, X_J) = \min_{i\in I}\min_{j\in J} D(X_i, X_j)\,.$$

Table 2.2 shows the values for the parameters α_{ij}, $\beta_{ii'}$, $\beta_{jj'}$, γ_{ij} and δ which determine the clustering method computed by formula (2.28). They are all gathered from the respective formulae (2.3), (2.5), (2.8), (2.9), (2.13), (2.16) and (2.19), derived in section 2.4.

2.7 Summary

Hierarchical clustering methods have been used since the beginning of the community detection problem. They are methods longly used by social scientists, and they consist in grouping individuals into groups of individuals or clusters that are both internally cohesive and externally isolated. In particular, agglomerative hierarchical clustering begins with each individual in a different singleton cluster, and successively merges clusters the most similar clusters at each step

until all individuals are in the same cluster. During this algorithmic process, a complete hierarchy of non-overlapping clusters is formed, providing a mesoscopic set of nested partitions of individuals into clusters corresponding to the possible cuts at different levels in the tree. We have started this chapter describing the classic pair-group agglomerative hierarchical clustering algorithm, and remembering the definition of valued tree, which is the output result of the pair-group agglomerative algorithm and it is graphically represented by a dendrogram.

However, when the pair-group algorithm is used and two or more distances between different clusters coincide during the amalgamation process, then agglomerative hierarchical methods suffer from the non-uniqueness problem, which consists in obtaining several hierarchical classifications from a unique set of tied proximity data. In such cases, selecting a unique classification can be misleading. This problem has traditionally been dealt with distinct criteria, which mostly consist on the selection of one out of several resulting hierarchies. Here we have proposed a new variable-group algorithm for agglomerative hierarchical clustering that solves the non-uniqueness problem. The output of this algorithm is a uniquely determined type of valued tree, that we have called a multivalued tree, and for which we have devised a new graphical representation called multidendrogram. In addition, we have illustrated the usefulness of our proposal with some results corresponding to data from a real example formed by the similarity values between twenty-three different soils. This example had been previously analyzed by other authors, detecting the existence of a tied value in the input matrix which was responsible for two different output hierarchies when using a pair-group approach. The use of our variable-group alternative leads to a unique result which coincides with the known classification of the soils data.

Afterwards, we have remembered the different definitions of distance between clusters for the most commonly used agglomerative hierarchical methods (single linkage, complete linkage, unweighted average, weighted average, unweighted centroid, weighted centroid, and joint between-within), and we have generalized them in order to use the variable-group algorithm. The use of any of these agglomerative methods implicitly specifies models for data and can provide misleading results of the class structure present in the data at study. We have reviewed the idea of space distortion in hierarchical clustering, which refers to strategies as space-conserving, space-contracting or space-dilating, and we have redefined these concepts in terms of the new variable-group approach. Finally, we have generalized Lance and Williams' formula, which enables us to obtain agglomerative hierarchical classifications in a recursive way, for the seven clustering methods studied in the chapter.

Although ties need not be present in the initial proximity data, they may arise during the agglomeration process. For this reason and given that the results of the variable-group method coincide with those of the pair-group method when there are no ties, we recommend to use directly the variable-group option. With a single action one knows whether ties exist or not, and additionally the subsequent solution is obtained.

Chapter 3

Financial Networks and Portfolios

Financial networks are an instance of correlation-based networks, that is, a particular type of complex systems where the nodes represent variables and each pair of nodes is connected by an edge with a weight related to the correlation coefficient between the two corresponding variables. The complexity of financial systems is reflected in their completely-connected networks, where all edges between nodes are present. In these cases, it is essential to be able to filter the networks into simpler relevant subnetworks. One possible filtering procedure is to use a *spanning tree* of the network, which is any subnetwork formed by the N nodes of the original network and a subset of $N-1$ edges keeping all the nodes connected. When the sum of the lengths in the subset of edges is minimum, the subnetwork is called a *minimum spanning tree*.

Minimum spanning trees have been recently used by Mantegna [60] to detect the taxonomy of a portfolio of stocks traded in a financial market, identifying hierarchical clusters of companies. In another paper by Bonanno *et al.* [16], the time evolution of stock indices was studied and significant changes in the world economy were identified using appropriate time horizons and a minimum spanning tree clustering method. In [52], the hierarchical structure explored by the minimum spanning tree seemed to give information about the influential power of the companies. And in [72], they also studied the minimum spanning tree as a strongly pruned representative of asset correlations and found it to be robust and descriptive of stock market events. In fact, all these works use minimum spanning trees as a way to obtain the subdominant ultrametric of the data and to detect the correspondent hierarchical structure of clusters. From this point of view, to use the minimum spanning tree is equivalent to use the single linkage hierarchical tree [77]. However, other ultrametrics are possible from other hierarchical clustering methods such as those described in chapter 2. The differences between using one clustering method or another are shown in section 3.1 of this chapter, and they are due to the different existing classifications in terms of space-distortion.

Another important application of hierarchical asset trees, in addition to their ability to form economically meaningful clusters, is in *portfolio selection*. In this problem, given a set of available securities or assets, one wants to find out the

optimum way of investing a particular amount of money in the assets. Many attempts have been made to solve this central problem, from the classical approach of Markowitz [61] to more sophisticated techniques [54, 71, 72, 75, 89], and in all these attempts correlations between asset prices play a crucial role. In [71, 72], for instance, the authors concentrate on the minimum spanning tree as a characteristic graph for the description of the correlations, and they study how the companies of the minimum risk Markowitz portfolio are located on the tree. In section 3.2 we give a detailed description of a novel portfolio selection approach, which reduces significantly the space and time complexity of the problem. The base idea is to minimize the risk of portfolios diversifying the investment into several assets belonging to different economic sectors. Hierarchical clustering trees can help us in the task of identifying distinct economic sectors, which will be probably represented by the main clusters in hierarchical trees.

The portfolio selection problem is an instance from the family of quadratic programming problems when the standard Markowitz mean-variance model is considered. However, this model has not got any cardinality constraint ensuring that every portfolio invests in a given number of different assets, neither uses any bounding constraint limiting the amount of money to be invested in each asset. This sort of constraints are very useful in practice. In order to overcome these inconveniences, the standard model can be generalized to include these constraints. With the latter model, the portfolio selection problem becomes a mixed quadratic and integer programming problem, and there is no exact algorithm able to solve this problem in an efficient way. Hence, the use of heuristic algorithms in this case is imperative. In the past some heuristic methods have been developed using Genetic Algorithms (GA) [18, 30, 56, 86, 94], Tabu Search (TS) [18, 82], and Simulated Annealing (SA) [18, 34, 51]. In section 3.3 we introduce a new heuristic method to solve the portfolio selection problem based on artificial Neural Networks (NN) [28], and we compare its results to those obtained using three representative methods based on GA, TS and SA.

3.1 Financial complex systems

3.1.1 Correlation-based networks

Correlation-based networks are a particular type of complex systems used to visualize the structure of pair correlations among a set of variables. Specifically, starting from a set of variables one can calculate the correlation coefficient between all possible pairs. If we identify the different variables with the nodes of the network, each pair of nodes can be thought to be connected by an edge with a weight related to the correlation coefficient between the two corresponding variables. Such a network is therefore completely connected. Well known examples of correlation-based networks are found in any portfolio of stocks traded in a financial market, by considering the synchronous evolution of the time series obtained from the daily differences between stock prices.

Let $P_i(t)$ be the closure price of stock i at day t. Then, the *return* of stock i at day t, $R_i(t)$, is defined as the logarithm of the ratio between two consecutive

closure prices:

$$R_i(t) = \ln\left(\frac{P_i(t)}{P_i(t-1)}\right) = \ln(P_i(t)) - \ln(P_i(t-1))\,. \tag{3.1}$$

The *mean* return of stock i is

$$\mu_i = \langle R_i \rangle\,, \tag{3.2}$$

where the brackets $\langle\cdot\rangle$ denote the arithmetic mean over the time interval of interest; and the *covariance* between returns of stocks i and j is the expectation value

$$\sigma_{ij} = \langle (R_i - \mu_i)(R_j - \mu_j)\rangle = \langle R_i R_j\rangle - \langle R_i\rangle\langle R_j\rangle\,. \tag{3.3}$$

Finally, in order to weight the edges of the financial complex network, we quantify the degree of similarity between two time series R_i and R_j by means of their *statistical correlation*, ρ_{ij}, computed over the investigated time period:

$$\rho_{ij} = \frac{\sigma_{ij}}{\sqrt{\sigma_{ii}\sigma_{jj}}} = \frac{\langle R_i R_j\rangle - \langle R_i\rangle\langle R_j\rangle}{\sqrt{(\langle R_i^2\rangle - \langle R_i\rangle^2)(\langle R_j^2\rangle - \langle R_j\rangle^2)}}\,. \tag{3.4}$$

Alternatively, one can assign distance values to the edges of a correlation-based network. This is the approach followed in [14, 39, 60], where the distance between two stocks i and j is measured as

$$d_{ij} = \sqrt{\langle(\tilde{R}_i - \tilde{R}_j)^2\rangle} = \sqrt{2(1-\rho_{ij})}\,, \tag{3.5}$$

with $\tilde{R}_i(t)$ being the standardized return of stock i at day t, that is,

$$\tilde{R}_i(t) = \frac{R_i(t) - \mu_i}{\sqrt{\sigma_{ii}}}\,. \tag{3.6}$$

The distance (3.5) measures the Euclidean distance between $\tilde{R}_i$ and $\tilde{R}_j$, and therefore it is a proper metric function ranging from 0 for perfectly correlated time series ($\rho_{ij} = +1$), to 2 for anticorrelated stocks ($\rho_{ij} = -1$).

3.1.2 Hierarchical asset trees

Portfolio theory suggests that, in order to minimize the risk involved in a financial investment, one should diversify among different assets by choosing those stocks whose price time evolutions are as diverse as possible. Therefore, it might be expected a connection between *hierarchical asset trees* (or multidendrograms) and the Markowitz portfolio selection scheme by means of the following rule: select stocks belonging to clusters that are as distant as possible from each other. This rule suggests the partitioning of a portfolio selection problem into several disjoint subproblems according to the main clusters of a hierarchical asset tree, which will probably represent different economic sectors. The risk is then divided into two components: one between economic sectors, that is minimized partitioning the problem according to the hierarchical clusters; and another one within sectors, that is minimized solving each subproblem separately. The time cost of this approach reduces significantly that of the original problem, not only because of the size reduction, but because of the possibility to solve the subproblems in parallel since they are disjoint.

Let us now follow this approach on real data from a stock market. In particular, we focus on the Standard and Poor's 500 (S&P 500) index, and we consider the two-years time period that goes from 31 December 2002 to 31 December 2004, both included, which means a total of 504 trading day intervals. From the data publicly available at *http://finance.yahoo.com/*, for each stock we have collected two different data: the daily volume, and the close price adjusted for dividends and splits. There is a set of 491 stocks with daily data for the whole time period under study, from which we have taken the subset of $N = 250$ stocks with highest mean volume. In this study we consider the time series of the returns from this subset of stocks, which gathers more than 87% of the volume traded by the set of 491 stocks.

The first thing that we study is the differences that arise in terms of space-distortion by the use of distinct clustering methods described in chapter 2: Single Linkage (SL), Complete Linkage (CL), Unweighted Average (UA), Weighted Average (WA), and Joint Between-Within (JBW)[1]. In figure 3.1 we show, in top-down increasing space-distortion order, the corresponding multidendrograms for the hierarchical clustering methods considered. We can see clearly how the Single Linkage method suffers from the *chaining effect*, and therefore it is not a good strategy in order to split the portfolio selection problem into subproblems of similar sizes. On the other hand, the multidendrograms obtained by means of both the Complete Linkage and the Joint Between-Within clustering methods show clear inner structures, corresponding to the branches of the hierarchical tree. However, before we can make any decision about which strategy to follow, we will do a further analysis of the data.

We take each hierarchical asset tree and we split it into disjoint subtrees following the top-down order. This means that, starting from the partition made up of a unique cluster with all the stocks in it, at each step we cut the highest branch remaining in the tree and we obtain a partition where clusters are identified as disjoint subtrees. In figure 3.2 we show the sizes of the biggest and smallest clusters in the partitions corresponding to the first 23 splittings, for each of the five hierarchical clustering methods under consideration. A first look at the results is enough to discard the Unweighted Average method, since it gives partitions similar to those of the Single Linkage method, i.e. partitions with clusters of sizes too dissimilar. Given that the test set has 250 stocks in it, we only plan to do the first three splittings of the hierarchical asset trees (partitions with 2, 3 and 4 clusters), otherwise we would get too small subproblems. Therefore, from the three remaining methods, we finally decide to discard the Weighted Average method because its partition into 4 clusters contains a cluster of size 2 which is not significant enough.

3.2 Portfolio selection

3.2.1 Formulation of the problem

First of all, as we introduce the notation that we are going to use in the rest of the chapter, let us remember the well known Markowitz mean-variance model [61] for the portfolio selection problem. Let N be the number of different assets, μ_i

[1]We do not consider here Centroid linkage methods, Weighted or Unweighted, because both of them can produce reversals and this means the loss of any hierarchical structure.

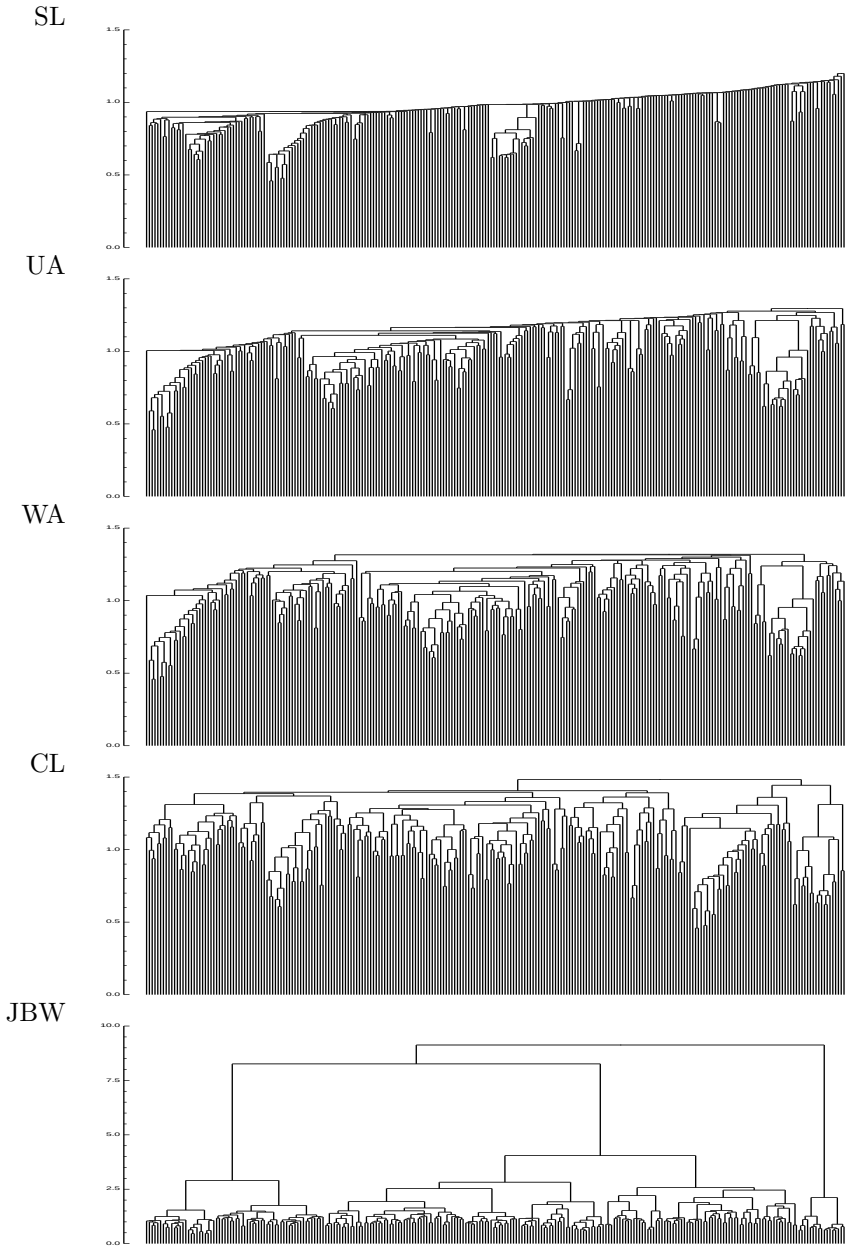


Figure 3.1: Hierarchical asset trees (multidendrograms) for the test set of $N = 250$ stocks belonging to the S&P 500 index. The trees are arranged in top-down increasing space-distortion order: Single Linkage (SL), Unweighted Average (UA), Weighted Average (WA), Complete Linkage (CL), and Joint Between-Within (JBW).

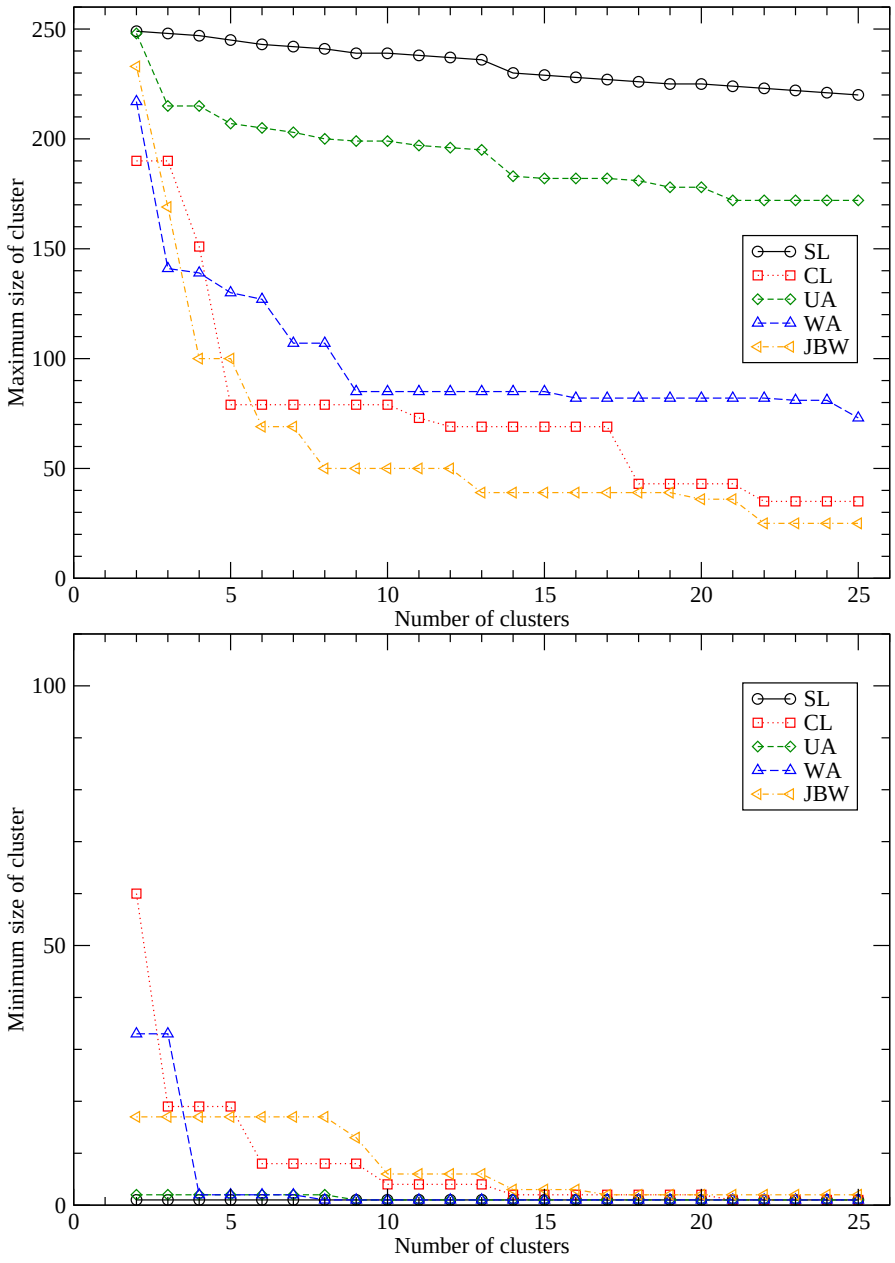


Figure 3.2: Maximum (top) and minimum (bottom) size of the clusters that appear in the first 23 divisions of the hierarchical asset trees. The number of clusters in these partitions ranges from 2 to 25.

the mean return of asset i, σ_{ij} the covariance between returns of assets i and j, and $\lambda \in [0,1]$ the risk aversion parameter. The decision variables x_i represent the proportion of capital to be invested in asset i. Using this notation, the standard Markowitz mean-variance model for the portfolio selection problem is:

$$\text{minimize} \qquad \lambda \left[\sum_{i=1}^{N} \sum_{j=1}^{N} x_i \sigma_{ij} x_j \right] + (1-\lambda) \left[-\sum_{i=1}^{N} \mu_i x_i \right], \tag{3.7}$$

$$\text{subject to} \qquad \sum_{i=1}^{N} x_i = 1\,, \tag{3.8}$$

$$0 \le x_i \le 1\,, \qquad i = 1, \ldots, N\,. \tag{3.9}$$

The case with $\lambda = 0$ represents maximizing the portfolio mean return (without considering the variance) and the optimal solution will be formed only by the asset with the greatest mean return. The case with $\lambda = 1$ represents minimizing the total variance associated to the portfolio (regardless of the mean returns) and the optimal solution will typically consist of several assets. Any value of λ inside the interval $(0,1)$ represents a tradeoff between mean return and variance, generating a solution between the two extremes $\lambda = 0$ and 1. Since every solution satisfying all the constraints (feasible solution) corresponds with one of the possible portfolios, from here on we will speak without distinguishing between solutions for the above problem and portfolios.

The portfolio selection problem is an instance of the family of multiobjective optimization problems. Therefore, one of the first things to do is to adopt a definition for the concept of optimality. Here we will use the Pareto optimality definition [81], that is, a feasible solution of the portfolio selection problem will be an *optimal* or *nondominated solution* if there is not any other feasible solution improving one objective without making worse the other.

Usually a multiobjective optimization problem has several optimal solutions. The objective function values of all these nondominated solutions form what it is called the *efficient frontier*. For the problem defined in equations (3.7)–(3.9), the efficient frontier is an increasing curve that gives the best tradeoff between mean return and variance (risk). In figure 3.3, we show an example of such a curve corresponding to the benchmark problem. This efficient frontier has been computed taking different values for the risk aversion parameter λ and solving exactly the corresponding portfolio selection problems. The objective function values of the resulting solutions give the points that form the curve in figure 3.3. We call this curve the *standard efficient frontier* in order to distinguish it from the *general efficient frontier*, corresponding to the general mean-variance portfolio selection model which will be described next.

With the purpose of generalizing the standard Markowitz model to include cardinality and bounding constraints, we will use a model formulation that can be also found in [18, 50, 82]. In addition to the previously defined variables, let K be the desired number of different assets in the portfolio with no null investment, ε_i and δ_i be respectively the lower and upper bounds for the proportion of capital to be invested in asset i, with $0 \le \varepsilon_i \le \delta_i \le 1$. The additional decision variables z_i are 1 if asset i is included in the portfolio and 0 otherwise. Then, the general

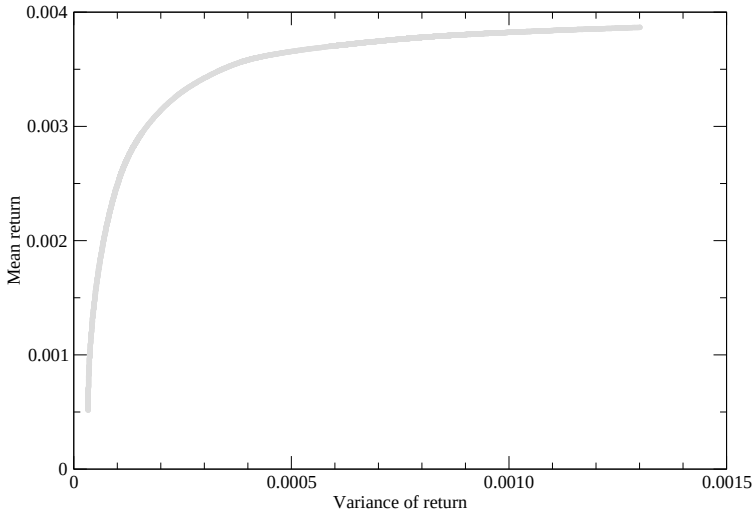


Figure 3.3: Standard efficient frontier corresponding to the S&P 500 benchmark data.

mean-variance model for the portfolio selection problem is:

$$\text{minimize} \qquad \lambda \left[\sum_{i=1}^{N} \sum_{j=1}^{N} x_i \sigma_{ij} x_j \right] + (1-\lambda) \left[- \sum_{i=1}^{N} \mu_i x_i \right] , \tag{3.10}$$

$$\text{subject to} \qquad \sum_{i=1}^{N} x_i = 1 \, , \tag{3.11}$$

$$\sum_{i=1}^{N} z_i = K \, , \tag{3.12}$$

$$\varepsilon_i z_i \leq x_i \leq \delta_i z_i \, , \qquad i = 1, \ldots, N \, , \tag{3.13}$$

$$z_i \in \{0, 1\} \, , \qquad i = 1, \ldots, N \, . \tag{3.14}$$

This formulation is a mixed quadratic and integer programming problem for which efficient algorithms do not exist. Another difference with the standard model is that in the presence of cardinality and bounding constraints the resulting efficient frontier, which we are going to call *general efficient frontier*, can be quite different from the one obtained with the standard mean-variance model. In particular, the general efficient frontier may be discontinuous [18, 50].

Figure 3.4 shows in gray the standard efficient frontier that solves the problem formulated in equations (3.7)–(3.9) for the S&P 500 benchmark data previously described. It also shows in black the general efficient frontier that solves the general problem in equations (3.10)–(3.14). The general efficient frontier has been computed using the values $K = 25$, $\varepsilon_i = 0.01$ and $\delta_i = 1$ for the problem formulation, and the values $\Delta\lambda = 0.02$ and $T = 1000N$ for the implementation of the Tabu Search algorithm in [18] used to solve this portfolio selection problem. Hence, we have tested 51 different values for the risk aversion parameter λ and the Tabu Search procedure has evaluated $1000N$ portfolios for each value of

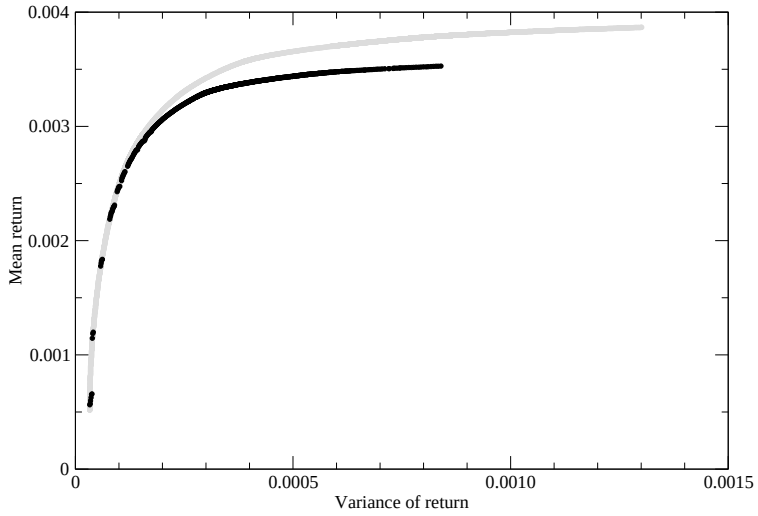


Figure 3.4: Standard (gray) and general (black) efficient frontiers corresponding to the S&P 500 benchmark data.

λ. Note that the general efficient frontier departs significantly from the standard efficient frontier in the zone corresponding to high-return solutions. This is due to the fact that high-return solutions are typically characterized by their low-diversified investments, whilst in the formulation of this benchmark problem we are searching for solutions with exactly $K = 25$ different investments.

Finally, in order to take advantage of the hierarchical clustering strategies previously analyzed, we need a way to divide a portfolio selection problem into several subproblems. The general mean-variance model shown in equations (3.10)–(3.14) can be divided into several subproblems according to any partition of assets into clusters. Thus, let N_c be the number of assets in cluster c and, without loss of generality, suppose that the N_c assets of cluster c are indexed consecutively from 1 to N_c. Let also $P_c = N_c/N$ be the proportion of assets in cluster c. Then, given a certain partition of assets into clusters, the portfolio selection problem can be divided into different subproblems, one for each cluster c, in the following way:

$$\text{minimize} \qquad \lambda \left[\sum_{i=1}^{N_c} \sum_{j=1}^{N_c} x_i \sigma_{ij} x_j \right] + (1-\lambda) \left[-\sum_{i=1}^{N_c} \mu_i x_i \right], \tag{3.15}$$

$$\text{subject to} \qquad \sum_{i=1}^{N_c} x_i = P_c \,, \tag{3.16}$$

$$\sum_{i=1}^{N_c} z_i = P_c K \,, \tag{3.17}$$

$$\varepsilon_i z_i \leq x_i \leq P_c \delta_i z_i \,, \qquad i = 1, \ldots, N_c \,, \tag{3.18}$$

$$z_i \in \{0, 1\} \,, \qquad i = 1, \ldots, N_c \,. \tag{3.19}$$

Once all the subproblems have been solved, the solutions forming the corre-

spondent efficient frontiers can be combined in order to get candidate solutions for the efficient frontier of the original problem. Note that the combination of solutions satisfying constraints (3.16)–(3.19) makes straightforward the satisfaction of constraints (3.11)–(3.14) by the combined solution. Therefore, after obtaining the candidate solutions by the combination process, it is only left to filter out the dominated solutions.

3.2.2 Results of the hierarchical strategies

Taking the sets of Pareto optimal portfolios obtained with the two hierarchical clustering strategies studied (Complete Linkage and Joint Between-Within), we can trace out the *hierarchical efficient frontiers* and compare them to the standard and general efficient frontiers. Doing so we get an upper bound of the error associated to each hierarchical strategy. We show these comparisons in figure 3.5, where the three different partitions are arranged by rows and the two hierarchical strategies are arranged by columns. As a reference, the standard and general efficient frontiers are drawn in lighter colors. For the partition into 2 clusters, the Joint Between-Within strategy covers a wider zone of the efficient frontier and therefore a greater variety of solutions are obtained using this strategy. On the contrary, for the partitions into 3 and 4 clusters, the Complete Linkage strategy covers better the efficient frontier. In the same figure, it can also be observed that the more the problem is divided into subproblems according to finer partitions, the more the solutions are gathered on the low-risk zone formed by portfolios with high diversification.

In table 3.1, we show some data describing the computational experiments. For each experiment performed we give two types of information: the cardinality of the clusters forming the correspondent partition, and the computation time needed to solve the problem by a parallel implementation of the algorithm. As it can be seen, the computation time clearly depends on the size of the biggest cluster in each partition, which is in total agreement with the conclusions of the theoretical study stating that the time cost function is linear with respect to the number of assets. Please note that the two sets of data shown in the first row are identical because they correspond to the same experiment, independently of the clustering strategy, which consists in considering all the assets put into a single cluster. These data are given just to have a reference allowing comparisons.

Finally, in order to analyze the relative quality of the solutions obtained with the two hierarchical strategies, for each division of the problem we have merged the corresponding pair of hierarchical efficient frontiers into a single one, and we have removed from it the dominated solutions. Then we have separated the resulting merged efficient frontier into the two parts that form it according to the hierarchical origin of the points, obtaining the results depicted in figure 3.6. According to these graphs we can conclude that the Complete Linkage method gives partitions of clusters which are generally better suited in order to divide and solve the portfolio selection problem using this hierarchical clustering approach.

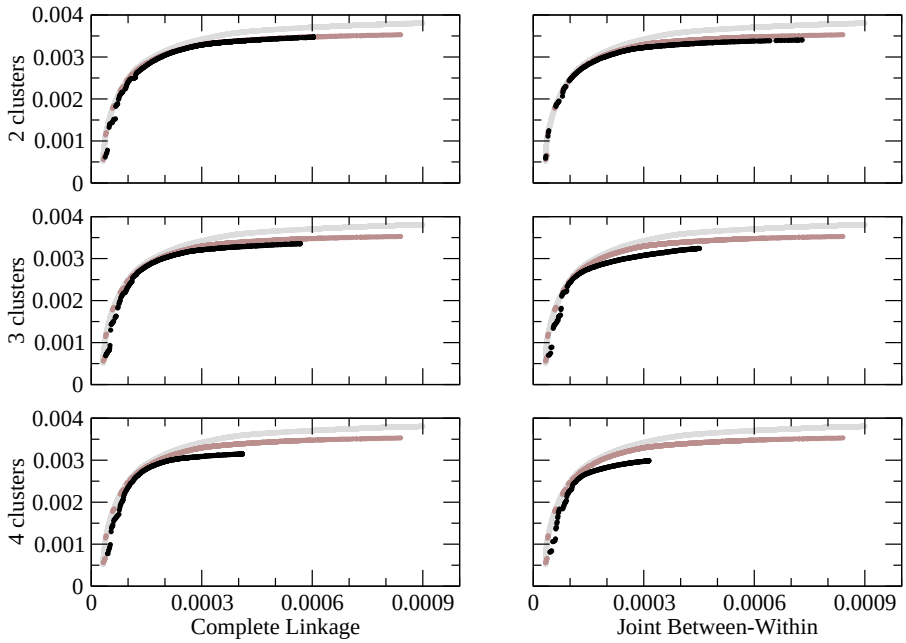


Figure 3.5: Hierarchical efficient frontiers for the S&P 500 benchmark data. The variance of return is represented on the x-axis and the mean return is represented on the y-axis. The standard and general efficient frontiers are drawn in lighter colors, whilst hierarchical efficient frontiers are drawn in black.

Table 3.1: Description of the computational experiments performed on the S&P 500 benchmark data considering two different hierarchical strategies. For each experiment, the cardinality of the clusters forming the correspondent partition and the computation time needed to solve the problem are given.

Clusters		Complete Linkage	Joint Between-Within
1	Partition	{250}	{250}
	Time	77 minutes	77 minutes
2	Partition	{190, 60}	{233, 17}
	Time	45 minutes	66 minutes
3	Partition	{190, 41, 19}	{169, 64, 17}
	Time	44 minutes	35 minutes
4	Partition	{151, 39, 41, 19}	{100, 69, 64, 17}
	Time	28 minutes	13 minutes

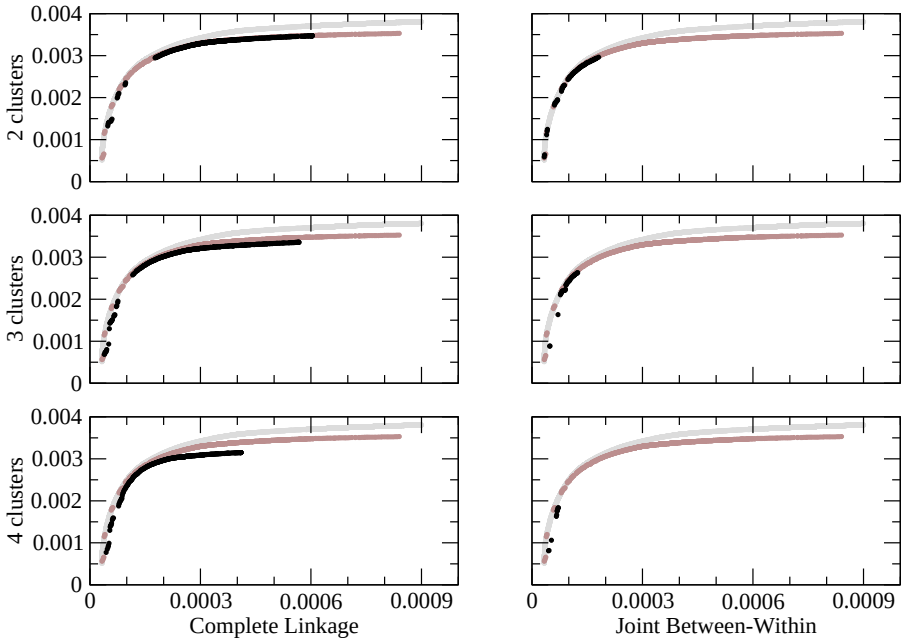


Figure 3.6: Contributions to merged efficient frontiers for the S&P 500 benchmark data. The variance of return is represented on the x-axis and the mean return is represented on the y-axis. The standard and general efficient frontiers are drawn in lighter colors, whilst contributions to merged efficient frontiers are drawn in black.

3.3 The Hopfield neural network

3.3.1 Energy function

There are two main approaches for solving combinatorial optimization problems using artificial neural networks (NN): Hopfield networks and Kohonen's self-organizing feature maps. While the latter are mainly used in Euclidean problems, the Hopfield networks have been widely applied in different classes of combinatorial optimization problems [83]. Although the problem at hand is not a combinatorial optimization one, we take advantage of the fact that the objective function in equation (3.10) has the same form than the energy function in Hopfield networks and, consequently, it will be minimized when we follow the Hopfield dynamics.

The Hopfield network [47] is an artificial NN model with a single layer of neurons fully connected, that is, all the neurons are connected to each other as well as to themselves. The N variables in the problem are represented by N neurons in the network and, given that we have defined x_i as the proportion of capital to be invested in asset i, the state of neuron i will be also represented by x_i. Using this notation, the energy function for the Hopfield network has the following appearance:

$$E(x) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} x_i w_{ij} x_j - \sum_{i=1}^{N} b_i x_i \,, \tag{3.20}$$

where b_i is the constant external input or *bias* for neuron i, and w_{ij} is the weight of the synaptic connection from neuron i to neuron j. Now, looking at the objective function of the portfolio selection problem,

$$f(x) = \lambda \left[\sum_{i=1}^{N}\sum_{j=1}^{N} x_i \sigma_{ij} x_j\right] + (1-\lambda)\left[-\sum_{i=1}^{N} \mu_i x_i\right] , \tag{3.21}$$

and just comparing it with the energy function in equation (3.20), we get the corresponding values for the synaptic weights

$$w_{ij} = -2\lambda\sigma_{ij} \,, \tag{3.22}$$

and for the external inputs

$$b_i = (1-\lambda)\mu_i \,. \tag{3.23}$$

3.3.2 Network dynamics

From here on, let us consider that we work with discrete time. Hence, if the state of neuron i at time t is represented by $x_i(t)$, the equations that govern the dynamics of the Hopfield network are:

$$x_i(t+1) = G_i(h_i(t)) \,, \qquad i = 1, \ldots, N \,, \tag{3.24}$$

where $h_i(t)$ is the input to neuron i at time t,

$$h_i(t) = \sum_{j=1}^{N} w_{ji} x_j(t) + b_i \,, \tag{3.25}$$

and G_i is the activation function, for which we use the sigmoid

$$G_i(h_i) = \varepsilon_i + \frac{\delta_i - \varepsilon_i}{1 + \exp(-\beta(h_i - \gamma))}\,, \tag{3.26}$$

with a gain $\beta > 0$ and a centering constant γ. In our simulations γ has been assigned the middle value between the maximum and the minimum initial inputs $h_i(0)$. ε_i and δ_i are used to ensure that the outputs of the sigmoid fall inside the interval $[\varepsilon_i, \delta_i]$, as it is required by the constraint in equation (3.13). Without loss of generality, let us suppose that all the lower bounds take the same value ($\varepsilon_i = \varepsilon$) and all the upper bounds are also identical ($\delta_i = \delta$). The output vector in a Hopfield network represents the solution for the problem at hand, and it lies inside the hypercube $[\varepsilon, \delta]^N$. The stability of the network can be proved defining the so called energy function for the network and proving that its time derivative is nonincreasing.

The nonlinear nature of the Hopfield network produces multiple equilibrium points. For any given set of initial conditions, $x(0)$, the symmetric Hopfield network with $w_{ji} = w_{ij}$ will converge towards a stable equilibrium point. When the network is deterministic, the position of that point is uniquely determined by the initial conditions: all the initial conditions that lie inside the attraction field of an equilibrium point will converge asymptotically towards that point. The number of equilibrium points and their exact positions are determined by the network parameters w_{ij} and β. When the gain β is small, the number of equilibrium points is low (possibly as low as 1) and they all lie inside the hypercube $[\varepsilon, \delta]^N$. However, as the gain increases, the number of equilibrium points also increases and their positions move towards the vertices of the hypercube. When the gain tends to its extreme values, $\beta \to +\infty$, the equilibrium points reach the hypercube vertices and are maximum in number. In this case, the energy function for the network has the appearance shown in equation (3.20).

In this work we update the neurons asynchronously, that is, only one neuron at a time. The neurons to be updated are selected randomly. This way of updating does not change the positions of the equilibrium points in the network, but it does change the descending path through the energy surface. Therefore, initial conditions that originally were attracted to a particular equilibrium point, can be attracted towards a different equilibrium point when using asynchronous updating.

For solving the portfolio optimization problem we have implemented a Hopfield network with gains β changing through time [74] and avoiding the saturation of the activation function G_i. The gains used are such that the image interval of the terms $\beta(h_i(0) - \gamma)$ in equation (3.26) widens linearly from $[-10, +10]$ to $[-20, +20]$. The initial gain values produce few equilibrium points and hence, regardless of the initial conditions, the network converges towards these points. As time passes, gains are linearly increased, producing energy surfaces with a higher number of equilibrium points and moving these equilibrium points towards the vertices of the hypercube $[\varepsilon, \delta]^N$.

Another problem that must be addressed is the possibility of convergence of the symmetric Hopfield network ($w_{ji} = w_{ij}$) to cycles of length 2. In order to avoid this undesired behavior in our network dynamics, the following discrete

model has been used:

$$x_i(t+1) = (1-\alpha_i)x_i(t) + \alpha_i G_i\left(\sum_{j=1}^{N} w_{ji}x_j(t) + b_i\right), \tag{3.27}$$

with $\alpha_i \in (0,1]$. In [91] it is shown that periodic points that are not fix points can appear, specially when all $\alpha_i = 1$. However, if synaptic weights are symmetric ($w_{ji} = w_{ij}$) and

$$w_{ii} > \frac{\alpha_i - 2}{\alpha_i \beta}, \tag{3.28}$$

then the above discrete model has the sequential dynamics convergent to fix points for any $\alpha_i \in (0,1]$. Since the synaptic weights w_{ii} are fixed from the beginning and the gains β are linearly increased, given any particular pair of values w_{ii} and β, what one must do here is to give a value to α_i satisfying the previous condition (3.28).

3.3.3 Constraints satisfaction

When solving any optimization problem using a Hopfield network, the problem constraints usually appear in the energy function. However, in our case this is not necessary. First, regarding the constraint $x_i \in [\varepsilon_i, \delta_i]$ in equation (3.13), we can say that it will be satisfied using as the activation function a sigmoid such as the one defined in equation (3.26), since its outputs already lie inside the desired interval.

In order to satisfy the cardinality constraint in equation (3.12), we begin our heuristic algorithm with a NN having $3K/2$ neurons that follow the already described Hopfield dynamics. Doing so we get a minimum for the objective function. Next thing to do is pruning the least representative neuron, that is, the one with the smallest output. Then we update this new network (with one less neuron) following the same Hopfield dynamics. These two steps, neuron pruning and objective function minimization, are repeated until the network has exactly K neurons. The remaining neurons are a solution for our original portfolio selection problem.

We are only left to consider the constraint in equation(3.11). To satisfy this constraint we evaluate the feasibility of every portfolio using the same greedy algorithm that has been used in [18], which changes the proportions of capital x_i to be invested in each selected asset in order to ensure, if possible, that all constraints are satisfied. In a first step, the algorithm assigns to all x_i corresponding to a selected asset its lower limit ε_i plus a fraction proportional to its current value. This ensures that all the constraints relating to the lower bounds are satisfied. In a second iterative step, the algorithm takes all the selected assets exceeding their respective upper limit δ_i and fixes them up to these upper limits. Then the rest of the selected assets that are not fixed up, are given a new value for x_i ensuring the lower bounds ε_i and adding a fraction of the free portfolio proportion. This iterative process is repeated until there is no asset out of its limits.

The only thing that we have changed in the greedy algorithm is the insertion of the current candidate solution into the set H with all the Pareto optimal solutions. In [18], the current solution is added to the set H only when it decreases the best objective function value found until that moment. Afterwards,

when the heuristic method finishes, all the dominated solutions are removed from H. However, this approach can leave out of this set solutions which are Pareto optimal. For example, let us consider the case with a risk aversion parameter $\lambda = 0.5$. If we first evaluate a solution a with variance of return equal to 0.001 and mean return equal to 0.005, then the objective function value for a is $f(a) = -0.002$. Now, if we evaluate in second place a solution b with variance of return 0.004 and mean return 0.006, the objective function value for b is $f(b) = -0.001$, which is greater than $f(a)$, so the solution b would not be added to the set H. However, in this case the two solutions should be included into H because both of them are Pareto optimal. What we do in our implementation of the greedy algorithm is to include in H all the evaluated solutions and, when the NN heuristic finishes, we remove from this set all the dominated solutions.

Bringing together all that we have said until now, we next show the NN heuristic used in this work:

```
function Neural_Network_Heuristic
  (Δλ: Float; {increment for the risk aversion parameter}
  T: Natural; {number of iterations}
  M: Natural) {number of portfolios in the set P}
  returns (H: Set_Of_Portfolio) is
var
  P: Set_Of_Portfolio; {set with candidate portfolios}
  P_Cand: Portfolio; {candidate portfolio}
begin
  H := Empty_Set();
  for λ := 0 to 1 by Δλ do
    P := Initialize_Portfolios_Randomly(M); {K assets in each portfolio}
    Evaluate_Portfolios(P, H); {greedy algorithm}
    for t := 1 to T by +1 do
      P_Cand := Select_Portfolio_Randomly(P);
      for k := 3*K/2 to K+1 by -1 do
        Follow_Hopfield_Dynamics(P_Cand); {P_Cand has k assets}
        Prune_Worst_Neuron(P_Cand);
      end for;
      Follow_Hopfield_Dynamics(P_Cand); {P_Cand has K assets}
      Evaluate_Portfolio(P_Cand, H); {greedy algorithm}
      Replace_Maximum_Portfolio(P_Cand, P);
    end for;
  end for;
  return H;
end Neural_Network_Heuristic;

procedure Follow_Hopfield_Dynamics
  (P_Cand: in out Portfolio) is
var
  R: Natural; {number of repetitions}
begin
  (γ, β) := Study_Initial_Inputs(P_Cand); {central input and gain value}
  for r := 1 to R by +1 do
    Update_Neuron(γ, β, P_Cand); {neuron to update selected randomly}
```

```
    Increase_Gain_Value(β);
  end for;
end Follow_Hopfield_Dynamics;
```

3.3.4 Results of the neural network heuristic

We have searched the general efficient frontier that solves the problem formulated in equations (3.10)–(3.14) for five sets of benchmark data that have been already used in [18, 50, 56, 82, 86]. These data correspond to weekly prices from March 1992 to September 1997 and they come from the indices: Hang Seng in Hong Kong, DAX 100 in Germany, FTSE 100 in UK, S&P 100 in USA and Nikkei 225 in Japan. The number N of different assets considered for each one of the test problems is 31, 85, 89, 98 and 225, respectively. The mean returns and covariances between returns have been calculated for the data. The sets of mean returns and covariances are publicly available at *http://people.brunel.ac.uk/~mastjjb/jeb/orlib/portinfo.html*.

All the results presented here have been computed using the values $K = 10$, $\varepsilon_i = 0.01$ and $\delta_i = 1$ for the problem formulation (as in reference [18]), and the values $\Delta\lambda = 0.02$, $T = 1000N$ and $M = 100$ for the implementation of the NN algorithm. Therefore, we have tested 51 different values for the risk aversion parameter λ and each one of the four heuristics has evaluated $1000N$ portfolios for each value of λ, without counting initializations.

The general efficient frontier has been computed using the former NN and our own implementation of three other heuristic algorithms presented in [18], which are based on GA, TS and SA. Our implementation of these additional heuristics uses the same parameter values than those presented in reference [18]. We would like to notice that the computational results presented here are not directly comparable to those presented in [18] due to the differences that exist at the moment of updating the set of Pareto optimal solutions (previously explained) and some other possible statistical fluctuations.

Taking the sets of Pareto optimal portfolios obtained with each heuristic we can trace out their heuristic efficient frontiers and compare them to the standard efficient frontiers. Doing so we get an upper bound of the error associated to each heuristic algorithm. We show these comparisons in figure 3.7, where the five problems are arranged by rows and the four heuristics are arranged by columns. Except for the first problem, where all four heuristics seem to obtain similar results, the four major problems show a common behavior. Looking at the portfolios with low mean return because of the low risk allowed (high values of the risk aversion parameter λ), we can see how the NN algorithm gets better results than the other three heuristics. On the contrary, the situation changes when we consider low values of λ and the increase of the mean return is the main objective, regardless of the risk. The solutions obtained in the first case consist of significant investments diversified in four or more of the $K = 10$ assets, whilst the solutions in the second case only have "significant investments"[2] in three or less of the $K = 10$ assets. Our explanation for the results obtained with the NN is that when the risk aversion parameter λ takes low values, the quadratic term in equation (3.21) decreases considerably and the objective function becomes almost linear, being no more a proper Hopfield energy function.

[2]Here we have applied the term "significant investment" to any investment above $1/K$.

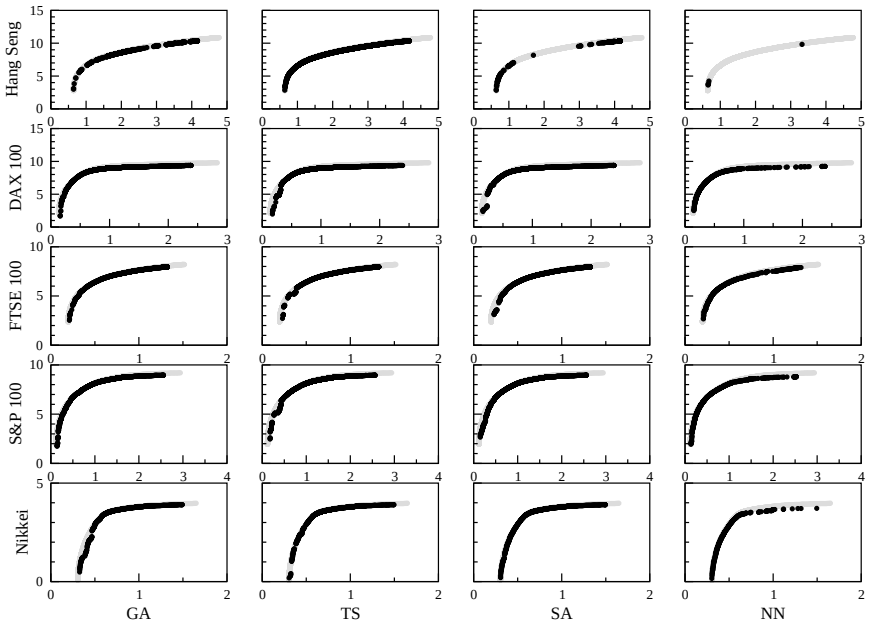


Figure 3.7: Heuristic efficient frontiers for five benchmark data. The variance of return ($\times 10^3$) is represented on the x-axis and the mean return ($\times 10^3$) is represented on the y-axis. Standard efficient frontiers are drawn in gray, whilst heuristic efficient frontiers are drawn in black.

Now let us take a look at some numerical results in table 3.2. First of all we have the number of points measure, which indicates how many of the portfolios evaluated by each heuristic method have persisted and finally appear in the corresponding heuristic efficient frontier. The results show that the highest numbers of points for all five problems come from TS, whilst the lowest numbers come from NN.

The number of points measure only gives an idea about the total number of solutions that appear in each heuristic efficient frontier, but it does not say anything about the quality of the solutions. Now, also in table 3.2, we present three error measures, one for variances, another one for mean returns and a third global error measure. Let the pair (v_i, r_i) represent the variance and mean return of a point in a heuristic efficient frontier. Let also $\hat{v}_i$ be the variance corresponding to r_i according to a linear interpolation in the standard efficient frontier. We define the variance of return error, φ_i, for any heuristic point (v_i, r_i) as the value $100(\hat{v}_i - v_i)/\hat{v}_i$ (note that this quantity will always be nonnegative). The average value of all the errors φ_i for the points in a heuristic efficient frontier gives us the variance of return error shown in table 3.2. In the same way, using the mean return $\hat{r}_i$ corresponding to v_i according to a linear interpolation in the standard efficient frontier, we define the mean return error, ψ_i, as the quantity $100(r_i - \hat{r}_i)/\hat{r}_i$. We get the average error for the mean returns computing the average of all the errors ψ_i, which appears in table 3.2 as the mean return error. There is not a single heuristic that gives better results than the others in any of these two average errors. Regarding the NN results we must say that, except for the second benchmark problem, they are worse than those obtained by the other heuristics. This is so because we are considering all possible values of the risk aversion parameter as a whole. In the next subsection we will see how the NN improves considerably its results when we consider only the best solutions of each heuristic.

We also give the results obtained with a third error measure defined in [18]. It is calculated averaging the minimums between the mean return errors, ψ_i, and the standard deviation of return errors, $\tilde{\varphi}_i$, which are similar to the variance of return errors but defined from the standard deviation of returns, s_i, i.e. $\tilde{\varphi}_i = 100(\hat{s}_i - s_i)/\hat{s}_i$. We present the values correspondent to the minimum error measure in table 3.2 to allow some kind of comparison between our results and those from reference [18]. However, we prefer to consider the variance of return error and the mean return error separately, because the use of the standard deviation for the calculation of the minimum error measure does not correspond exactly with the objective function of the problem at hand.

With regard to the computation times, the TS and the SA are the most efficient algorithms, followed by the GA and finally by the NN. Anyway, all four time cost functions are linear with respect to the number of assets N since the four algorithms have evaluated $1000N$ portfolios for each value of the risk aversion parameter λ, and the rest of operations in the algorithms (except for the crossover procedure in the GA) do not depend on N. The times presented in table 3.2 agree with it.

3.3.5 Merge process

In order to improve the results obtained separately by the four heuristic algorithms, we have merged the four heuristic efficient frontiers into a single one

Table 3.2: Numerical results for five benchmark problems. For each benchmark problem and each heuristic method we give: the number of points in the heuristic efficient frontier, three error measures (variance of return, mean return and minimum), and the computation time.

Index		GA	TS	SA	NN
Hang Seng	Number of points	3402	3659	2640	1108
	Variance error	3.9576	3.9329	3.8689	4.1039
	Mean error	1.1926	1.1500	1.1574	1.4530
	Minimum error	1.1321	1.1237	1.1203	1.2316
	Time (s)	47	16	18	390
DAX 100	Number of points	1828	2292	1264	573
	Variance error	26.1240	24.1340	26.8588	12.5914
	Mean error	2.6202	2.8490	2.6893	2.2060
	Minimum error	2.4457	2.6668	2.3896	1.5776
	Time (s)	162	45	47	1069
FTSE 100	Number of points	1284	1295	1267	426
	Variance error	3.3464	3.1458	3.6930	4.4663
	Mean error	0.9300	0.8954	1.3127	1.9636
	Minimum error	0.7310	0.7357	0.9512	1.2513
	Time (s)	160	51	60	1106
S&P 100	Number of points	1780	2318	1779	750
	Variance error	7.2039	7.6219	8.1602	8.3811
	Mean error	1.6130	1.4249	1.9672	2.6816
	Minimum error	1.3236	1.3130	1.7251	1.7922
	Time (s)	178	50	52	1211
Nikkei	Number of points	807	1027	984	312
	Variance error	4.9877	3.5724	3.4830	6.5924
	Mean error	3.3931	1.1581	1.2144	3.1050
	Minimum error	1.1415	0.5510	0.5458	1.4737
	Time (s)	570	120	121	2827

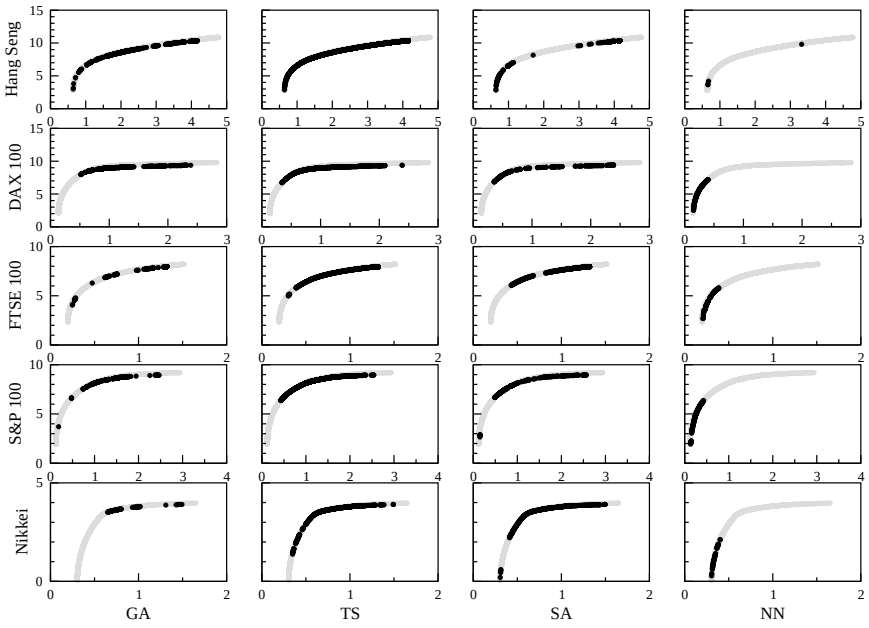


Figure 3.8: Contributions to merged efficient frontiers for five benchmark data. The variance of return ($\times 10^3$) is represented on the x-axis and the mean return ($\times 10^3$) is represented on the y-axis. Standard efficient frontiers are drawn in gray, whilst contributions to merged efficient frontiers are drawn in black.

and we have removed from it the dominated solutions. Then we have separated the resulting merged efficient frontier into the four parts that form it according to the heuristic origin of the points, getting the results shown in figure 3.8. Observe that these graphs confirm what we had already noticed in figure 3.7, which is that the NN gives better results than the other heuristics, except for the smallest benchmark problem, when we consider high values of the risk aversion parameter λ and, on the contrary, it gives worse solutions for low values of λ.

Continuing with the results of the merge process, let us now use the new merged efficient frontiers to compare the quality of the initial solutions provided by each heuristic method. We take the initial numbers of points in each heuristic efficient frontier (see table 3.2) and we compare them with the final numbers of points in the merged efficient frontiers that come from the corresponding heuristic, to obtain the percentage of points surviving the merge process which are shown in table 3.3. The initial quality of TS solutions is outstanding given that more than 90% of the portfolios provided by this algorithm survive the merge process. Next, we show the contribution of each heuristic to the merged efficient frontiers. As it can be seen, more than half of the points in the merged efficient frontiers come from the TS. Nevertheless, we would like to remember that most of these points correspond to low diversification portfolios.

Also in table 3.3, we give the values obtained with the three previously defined error measures when they are applied to the merged efficient frontiers. Limiting the error calculations only to the solutions that pass the merge process, we observe that the NN gets the lowest variance of return errors in the five

Table 3.3: Numerical results for five benchmark problems after the merge process. For each problem and each heuristic method we give: the number and percentage of points surviving the merge process, the contribution percentage, and three error measures (variance of return, mean return and minimum).

Index		GA	TS	SA	NN
Hang Seng	Nb points	1015	3340	270	5
	Points %	29.8	91.3	10.2	0.5
	Contribution %	21.9	72.1	5.8	0.1
	Variance error	4.4758	3.9046	4.5314	1.2279
	Mean error	1.2438	1.1431	1.1834	1.5304
	Minimum error	1.2321	1.1200	1.1167	0.3751
DAX 100	Nb points	635	2088	291	222
	Points %	34.7	91.1	23.0	38.7
	% contribucin	19.6	64.5	9.0	6.9
	Variance error	31.1493	23.2013	32.9453	2.3004
	Mean error	2.6802	2.3492	1.8412	2.7545
	Minimum error	2.6802	2.3492	1.8412	1.1326
FTSE 100	Nb points	220	1207	666	146
	Points %	17.1	93.2	52.6	34.3
	Contribution %	9.8	53.9	29.7	6.5
	Variance error	5.1662	2.7487	3.8031	2.5449
	Mean error	1.1129	0.5192	0.6501	3.1171
	Minimum error	0.9700	0.5160	0.6501	1.2544
S&P 100	Nb points	370	2146	456	287
	Points %	20.8	92.6	25.6	38.3
	Contribution %	11.4	65.8	14.0	8.8
	Variance error	6.6434	6.3510	7.3497	5.9020
	Mean error	0.8449	0.7978	0.9405	4.7956
	Minimum error	0.8349	0.7978	0.7381	2.7908
Nikkei	Nb points	122	943	648	53
	Points %	15.1	91.8	65.9	17.0
	Contribution %	6.9	53.4	36.7	3.0
	Variance error	4.6384	3.3516	3.6212	0.7172
	Mean error	0.6399	0.5624	0.6153	4.6097
	Minimum error	0.6399	0.5237	0.5002	0.3578

benchmark problems, being especially good the result obtained in the second problem. With respect to the mean return error, the situation is the opposite one and the NN gives the highest error values. Finally notice that in the GA, TS and SA columns of the same table there are several cases where the mean return error coincides with the minimum error. In these cases, the heuristics have the mean return errors always lower than the standard deviation errors.

3.4 Summary

Considering the synchronous evolution of the time series obtained from the daily differences between stock prices traded in a financial market, and identifying the different time variables with the nodes of a network, each pair of nodes can be thought to be connected by an edge with a weight related to the correlation coefficient between the corresponding pair of variables. In this chapter we have described the complexity of such correlation-based financial systems, reflected in their associated networks which result in completely-connected structures. In these cases, sometimes it is necessary to filter the networks into simpler relevant subnetworks. Such a filtering can be done, for instance, using hierarchical asset trees. In order to analyze the space-distortion differences that exist between the output asset trees of distinct hierarchical clustering methods, we have used a set of real data from the Standard and Poor's 500 index. We have only considered agglomerative methods that do not produce reversals in their output hierarchical trees, and we have arranged the results in space-distortion order. At one end of this classification appears the single linkage method, which suffers from the known chaining effect, whilst at the other end of the arrangement one finds the complete linkage and the joint between-within methods, both of them showing clear inner structures corresponding to the branches of their hierarchical trees.

Then, we have focused on solving the portfolio selection problem and tracing out its efficient frontier. Instead of using the standard Markowitz mean-variance model, we have used a generalization of it that includes cardinality and bounding constraints. Dealing with this type of constraints, the portfolio selection problem becomes a mixed quadratic and integer programming problem for which no computational efficient algorithms are known. First, we have used the hierarchical asset trees to device a new portfolio selection approach based on the division of the problem into several subproblems according to the clusters arising in the asset trees. For doing that, we have used the two more promising hierarchical methods for the problem at hand, namely complete linkage and joint between-within.

In a second approach, we have developed a heuristic method based on the Hopfield Neural Network. First, we have remembered the peculiar form of the Hopfield energy function, which resembles the objective function of the portfolio selection problem. Next, we have described the network dynamics and we have analyzed how to satisfy each constraint of the problem. Finally, we have taken the results obtained with our Neural Network and we have compared them to those obtained using three other heuristic methods based on Genetic Algorithms, Tabu Search and Simulated Annealing.

Chapter 4

Size Reduction of Complex Networks

The study of the modular structure in real-world complex networks has become a classical subject in the area because several aspects of the problem are both challenging and interesting. The challenge comes from the difficulty for unveiling the best partition of the network in terms of communities, in the sense of groups of nodes that are more intraconnected rather than interconnected between them [35]. The interest comes from the fact that this level of description could help to understand the interplay between network topology and functionality [40, 43], and also because it resembles the coarse graining process in statistical physics to describe systems at the mesoscale.

Hierarchical clustering methods have the advantage of not requiring previous knowledge about the number and size of communities in a network. The output of the algorithm has the form of a tree representing a complete hierarchy of possible partitions into communities for the network, and it could be desirable to know which of those partitions describes in a better way the community structure of the network, i.e. where should a multidendrogram be cut off. With that purpose we look for an appropriate *quality function*, that is, a quantitative criterion to evaluate how good partitions are. Several interesting proposals have been taken from the field of statistical physics, such as those based on spin models. One of the best known quality functions is the *modularity* defined by Newman and Girvan [69], and in section 4.1 we describe modularity under a unified framework of quality functions coming from a particular spin glass model.

Modularity optimization has become one of the best approaches to detect community structure in networks. The reason is simple: if modularity is a good quality function, then partitions with high modularity values should be among the best partitions. However, modularity optimization cannot be performed by exhaustive search since the number of different partitions for a given network with N nodes is equal to the Bell number for N [13], which grows at least exponentially. Indeed, the computational complexity of modularity optimization is in the NP (Non-deterministic Polynomial-time) class [17], therefore it is quite improbable to find a solution within a time growing polynomially with the size of the network. Several authors have attacked the problem proposing

different heuristic algorithms that are able to find, within reasonable computational time, quite good approximations to the partitions of maximum modularity [19, 25, 40, 66, 68, 76]. Nevertheless, when facing the decomposition into communities of very large networks, optimality is usually sacrificed in favor of computational time. In this chapter, we demonstrate that it is possible to reduce the size of complex networks while preserving the value of modularity, independently of the partition under consideration [5]. In section 4.2 we describe the property of modularity that allows the size reductions of some complex networks, and in section 4.3 we proof certain possible analytic reductions preserving modularity. Finally, in section 4.4 we estimate the amount of size reduction that one might expect in terms of the degree distribution, and we show some size reductions obtained for several real complex networks.

4.1 Quality functions

Communities in complex networks are usually understood as groups of nodes that are densely connected between them and only sparsely connected with the rest of the network. Reichardt and Bornholdt [79] mapped the community detection problem onto finding the ground state of an infinite range Potts spin glass model, where similarity measures are translated into coupling strengths. They argued that any quality function for an assignment of nodes into communities should follow this simple principle: group together what is linked, and keep apart what is not. From here, two requirements for such a quality function arise: it should i) reward internal links between nodes in the same community (in the same spin state), and ii) penalize missing links between nodes in the same community. This leads to the following Hamiltonian representing the energy of the system:

$$\mathcal{H} = -\sum_i \sum_j r_{ij} A_{ij} \delta(C_i, C_j) + \sum_i \sum_j s_{ij} (1 - A_{ij}) \delta(C_i, C_j)\,, \tag{4.1}$$

where A_{ij} denotes the adjacency matrix of the graph, with $A_{ij} = 1$ if an edge exists and 0 otherwise; C_i and C_j denote the respective community indices (or spin states) of nodes i and j, with the Kronecker delta function, $\delta(C_i, C_j)$, taking the value 1 when i and j are in the same community, 0 otherwise; and r_{ij} and s_{ij} denote the weights of the contributions corresponding to the existent and missing links, respectively.

A convenient choice of weights r_{ij} and s_{ij}, such that the contribution of existent and missing links can be adjusted through a unique parameter γ, is:

$$r_{ij} = 1 - \gamma p_{ij}\,, \tag{4.2}$$

$$s_{ij} = \gamma p_{ij}\,, \tag{4.3}$$

where p_{ij} denotes the probability that a link exists between nodes i and j, normalized such that $\sum_i \sum_j p_{ij} = 2m$ (this means that m is the total number of links in the graph). For $\gamma = 1$, this leads to the natural situation where the total amount of energy that can be contributed by existent and missing links is equal:

$$\sum_i \sum_j r_{ij} A_{ij} = \sum_i \sum_j s_{ij} (1 - A_{ij})\,. \tag{4.4}$$

Henceforth, we will consider that the parameter γ takes a constant value equal to 1. The choice of the weights r_{ij} and s_{ij} in (4.3), and the latter consideration about the value of the parameter γ, allow us to further simplify the Hamiltonian

$$\mathcal{H} = -\sum_i \sum_j (A_{ij} - p_{ij})\delta(C_i, C_j)\,. \tag{4.5}$$

This represents a spin glass with couplings $J_{ij} = A_{ij} - p_{ij}$ between all pairs of nodes: ferromagnetic where links between nodes are present, and antiferromagnetic where links are absent.

The Hamiltonian in equation (4.5) compares the true distribution of links in the graph with the expected distribution given by a particular null model of connectivity p_{ij}. Depending on the problem under study, one can assume different expressions for p_{ij}, which allows for the comparison of the quality function for graphs with different topology. One possible choice for the link distribution model p_{ij} may take into account that the network exhibits a particular degree distribution. Since links are more probable between nodes of high degree, links between these nodes get a higher weight. Taking this fact into account, we can consider

$$p_{ij} = \frac{k_i k_j}{2m}\,, \tag{4.6}$$

where k_i and k_j are the respective degrees of nodes i and j (i.e. the number of edges attached to them).

In order to measure how good a community structure found by an algorithm is, many authors have given values of a quality function defined by Newman and Girvan as *modularity* [69], and which can be expressed as:

$$Q = \frac{1}{2m} \sum_i \sum_j \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(C_i, C_j)\,. \tag{4.7}$$

This already resembles the Hamiltonian of equation (4.5) when p_{ij} takes the form in equation (4.6). It is now clear that we can write

$$Q = -\frac{1}{2m}\mathcal{H}\,. \tag{4.8}$$

Therefore, maximum modularity is reached when the Hamiltonian is minimal. To maximize the modularity of a community structure is hence equivalent to finding the spin configuration that minimizes this Hamiltonian.

In terms of the weighted adjacency matrix, w_{ij}, that represents the value of the weight of the link between nodes i and j, the definition of modularity is expressed as [64]:

$$Q = \frac{1}{2w} \sum_i \sum_j \left(w_{ij} - \frac{w_i w_j}{2w} \right) \delta(C_i, C_j)\,, \tag{4.9}$$

where the strength of node i is $w_i = \sum_j w_{ij}$, and the total strength of the network is $2w = \sum_i w_i = \sum_i \sum_j w_{ij}$. Given a network partitioned into communities, the modularity measures the fraction of edges in the network falling within communities, minus the expected value of the same function in a null case network with the same nodes and the same partition into communities,

but with edges redistributed at random preserving the strength of the nodes. In general, random networks are not expected to exhibit modular structure beyond certain fluctuations. When the number of edges located inside communities is similar to the expected number of random edges, then modularity values are in proximity to zero. The higher the modularity value, the better the partitioning into communities is because more deviates from the null case.

The definition of modularity can be also extended, preserving its semantics in terms of probability, to the scenario of directed networks as follows:

$$Q = \frac{1}{2w} \sum_i \sum_j \left(w_{ij} - \frac{w_i^{\text{out}} w_j^{\text{in}}}{2w} \right) \delta(C_i, C_j) \,, \tag{4.10}$$

where w_i^{out} and w_j^{in} are respectively the output and input strengths of nodes i and j,

$$w_i^{\text{out}} = \sum_j w_{ij} \,, \tag{4.11}$$

$$w_j^{\text{in}} = \sum_i w_{ij} \,, \tag{4.12}$$

and the total strength of the network is

$$2w = \sum_i w_i^{\text{out}} = \sum_j w_j^{\text{in}} = \sum_i \sum_j w_{ij} \,. \tag{4.13}$$

When the network is undirected, then the output and input strengths are equal ($w_i^{\text{out}} = w_i^{\text{in}} = w_i$), thus recovering the standard definition of strength.

4.2 Network reduction preserving modularity

4.2.1 Reduced network

Let G be a weighted complex network of size N, with weights $w_{ij} \geq 0$. If the network is unweighted, the weights matrix becomes the usual connectivity matrix, with values equal to 1 for connected pairs of nodes, and 0 otherwise. We will assume that the network may be directed, i.e. represented by a non-symmetric weights matrix.

Any grouping of the N nodes of the complex network G in N' parts may be represented by a function $R : \{1, \ldots, N\} \longrightarrow \{1, \ldots, N'\}$ which assigns a group index $R_i \equiv R(i)$ to every i-th node in G. The *reduced network* G' in which each of these groups is replaced by a single node may be easily defined in the following way: the weight w'_{rs} between the nodes which represent groups r and s is the sum of all the weights connecting vertices in these groups,

$$w'_{rs} = \sum_i \sum_j w_{ij} \delta(R_i, r) \delta(R_j, s) \,, \qquad r, s \in \{1, \ldots, N'\} \,. \tag{4.14}$$

For unweighted networks the value of w'_{rs} is just the number of arcs from the first to the second group of nodes. It must be emphasized that a node r of the reduced network G' acquires a *self-loop* if $w'_{rr} \neq 0$, which summarizes the internal connectivity of the nodes of G forming this group.

The output and input strengths of the reduced network G' are:

$$w'^{\text{out}}_r = \sum_s w'_{rs} = \sum_i \sum_j w_{ij}\delta(R_i, r) \sum_s \delta(R_j, s) = \sum_i w_i^{\text{out}}\delta(R_i, r)\,, \quad (4.15)$$

$$w'^{\text{in}}_s = \sum_r w'_{rs} = \sum_j \sum_i w_{ij}\delta(R_j, s) \sum_r \delta(R_i, r) = \sum_j w_j^{\text{in}}\delta(R_j, s)\,, \quad (4.16)$$

and its total strength, $2w'$, is equal to the total strength $2w$ of the original network:

$$2w' = \sum_r w'^{\text{out}}_r = \sum_s w'^{\text{in}}_s = \sum_i w_i^{\text{out}} = \sum_j w_j^{\text{in}} = 2w\,. \quad (4.17)$$

4.2.2 Modularity preservation

The main property of the reduced network is the preservation of modularity (4.9) or (4.10), i.e. the modularity of any partition of the reduced network is equal to the modularity of its corresponding partition of the original network.

More precisely, let $C' : \{1, \ldots, N'\} \longrightarrow \{1, \ldots, M\}$ be a partition in M clusters of the reduced network G'. Its corresponding partition $C : \{1, \ldots, N\} \longrightarrow \{1, \ldots, M\}$ of the original graph is given by the composition of the reducing function R with the partition C', i.e. $C = C' \circ R$. Therefore, the statement of the previous paragraph becomes:

$$Q'(C') = Q(C)\,. \quad (4.18)$$

The proof is straightforward:

$$\begin{aligned}
Q'(C') &= \frac{1}{2w'} \sum_r \sum_s \left(w'_{rs} - \frac{w'^{\text{out}}_r w'^{\text{in}}_s}{2w'} \right) \delta(C'_r, C'_s) \\
&= \frac{1}{2w} \sum_r \sum_s \left(\sum_i \sum_j w_{ij}\delta(R_i, r)\delta(R_j, s) \right. \\
&\quad \left. - \frac{1}{2w} \sum_i w_i^{\text{out}}\delta(R_i, r) \sum_j w_j^{\text{in}}\delta(R_j, s) \right) \delta(C'_r, C'_s) \\
&= \frac{1}{2w} \sum_i \sum_j \left(w_{ij} - \frac{w_i^{\text{out}} w_j^{\text{in}}}{2w} \right) \sum_r \sum_s \delta(R_i, r)\delta(R_j, s)\delta(C'_r, C'_s) \\
&= \frac{1}{2w} \sum_i \sum_j \left(w_{ij} - \frac{w_i^{\text{out}} w_j^{\text{in}}}{2w} \right) \delta(C'_{R_i}, C'_{R_j}) \\
&= \frac{1}{2w} \sum_i \sum_j \left(w_{ij} - \frac{w_i^{\text{out}} w_j^{\text{in}}}{2w} \right) \delta(C_i, C_j) \\
&= Q(C)\,. \quad (4.19)
\end{aligned}$$

We have found a relevant property of modularity, namely that those nodes forming a community in the optimal partition can be represented by a unique node in the reduced network. Each node in the reduced network summarizes

the information necessary for the calculation of modularity in its self-loop (that accounts for the intraconnectivity of the community) and its arcs (that account for the total strengths with the rest of the network). The question now is: how do we determine which nodes will belong to the same community in the optimal partition, before this partition is obtained? The answer to this question will provide us with a size reduction method in complex networks preserving modularity.

4.3 Analytic reductions

Here we give the proof for certain possible analytic size reductions of weighted networks, undirected and directed.

4.3.1 Reductions for undirected networks

The modularity of an undirected network may be written as

$$Q = \sum_i q_i \,, \tag{4.20}$$

where

$$q_i = \frac{1}{2w} \sum_j \left(w_{ij} - \frac{w_i w_j}{2w} \right) \delta(C_i, C_j) \tag{4.21}$$

is the contribution to modularity of the i-th node. If we allow this node to change of community, the value of C_i becomes a parameter and therefore it is useful to define

$$q_{i,r} = \frac{1}{2w} \sum_j \left(w_{ij} - \frac{w_i w_j}{2w} \right) \delta(C_j, r) \,, \qquad q_i = q_{i,C_i} \,, \tag{4.22}$$

which accounts for the contribution of the i-th node to modularity if it were in community r. The separation of the self-loop term, which does not depend on which community node i belongs to, yields to the definition of

$$\tilde{q}_{i,r} = \frac{1}{2w} \sum_{j \neq i} \left(w_{ij} - \frac{w_i w_j}{2w} \right) \delta(C_j, r) \,, \qquad \tilde{q}_i = \tilde{q}_{i,C_i} \,, \tag{4.23}$$

and

$$\tilde{Q} = \sum_i \tilde{q}_i = \frac{1}{2w} \sum_i \sum_{j \neq i} \left(w_{ij} - \frac{w_i w_j}{2w} \right) \delta(C_j, C_i) \,, \tag{4.24}$$

satisfying

$$q_{i,r} = \tilde{q}_{i,r} + \frac{1}{2w} \left(w_{ii} - \frac{w_i^2}{2w} \right) \tag{4.25}$$

and

$$Q = \tilde{Q} + \frac{1}{2w} \sum_i \left(w_{ii} - \frac{w_i^2}{2w} \right) \,. \tag{4.26}$$

The role of these individual node contributions to modularity becomes evident in the expression of the change of modularity when node i goes from community r to community s:

$$\Delta Q = 2(\tilde{q}_{i,s} - \tilde{q}_{i,r}) \,. \tag{4.27}$$

As a particular case, a node i that forms its own community, i.e. an isolated node, which moves to any community s produces a change in modularity

$$\Delta Q = 2\tilde{q}_{i,s}\,. \tag{4.28}$$

Therefore, if there exists a community s for which $\tilde{q}_{i,s} > 0$, then node i cannot be isolated in the partition of optimal modularity. This existence is easily proved by considering the sum of $\tilde{q}_{i,r}$ for all communities:

$$\begin{aligned}\sum_r \tilde{q}_{i,r} &= \frac{1}{2w}\sum_{j\neq i}\left(w_{ij} - \frac{w_i w_j}{2w}\right)\sum_r \delta(C_j, r)\\ &= \frac{1}{2w}\sum_{j\neq i}\left(w_{ij} - \frac{w_i w_j}{2w}\right)\\ &= -\frac{1}{2w}\left(w_{ii} - \frac{w_i^2}{2w}\right)\,,\end{aligned} \tag{4.29}$$

where we have made use of the definitions of strength w_i and total strength $2w$ for the simplification of the expression. Thus,

$$w_{ii} \leq \frac{w_i^2}{2w} \quad\Rightarrow\quad \sum_r \tilde{q}_{i,r} \geq 0 \quad\Rightarrow\quad \exists s : \tilde{q}_{i,s} \geq 0\,, \tag{4.30}$$

completing the proof that there are no isolated nodes in the configuration which maximizes modularity, unless they have a big enough self-loop.

Now, it remains the problem of the determination of a node j, an acquaintance of node i in its optimal community, in order to group them ($R_i = R_j$) in a single equivalent node with a self-loop, as explained above. If we know that nodes i and j share the same community at maximum modularity, the reduced network will be equivalent to the original one as regards modularity: no information lost, and a smaller size. Taking into account that the sign of the $\tilde{q}_{i,r}$ can only be positive if there is a link between node i and another node in community r, the only candidates to be the right acquaintance of any node are its neighbors in the network.

The simplest particular cases are *hairs*, i.e. nodes connected to the network with only one link. Hence, a hair can be analytically grouped with its neighbor k if

$$w_{ii} \leq \frac{w_i^2}{2w}\,, \tag{4.31}$$

producing a self-loop for node k of value

$$w'_{kk} = w_{ii} + 2w_{ik}\,. \tag{4.32}$$

When node i has no self-loop ($w_{ii} = 0$), this condition is always fulfilled (see figure 4.1a). Note also that in the particular case of unweighted undirected networks, the recursive process of reducing hairs allows only one iteration, because after that new hairs will have self-loops that will not satisfy (4.31).

Another solvable structure is the *triangular hair*, in which two nodes i and j have only one link connecting them, two more links from i and j to a third node k, and possibly self-loops. In this case, if

$$w_{ii} \leq \frac{w_i^2}{2w} \qquad \text{and} \qquad w_{jj} \leq \frac{w_j^2}{2w}\,, \tag{4.33}$$

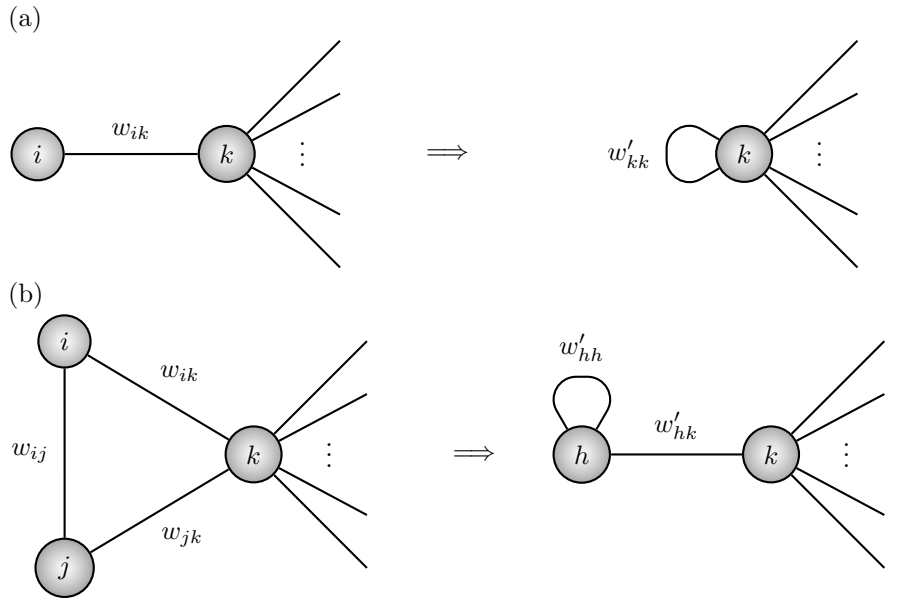


Figure 4.1: Analytic reductions for undirected networks: (a) example of a *hair* reduction; (b) example of a *triangular hair* reduction. The widespread case of unweighted networks, all weights equal to 1, implies that in the reduction (a), $w'_{kk} = 2$, and in the reduction (b), $w'_{hh} = 2$ and $w'_{hk} = 2$.

nodes i and j share the same community in the optimal partition and therefore may be grouped as a single node h. Moreover, the resulting structure becomes a simple hair, which can be grouped with node k if

$$w'_{hh} \leq \frac{w_h'^2}{2w'}\,, \tag{4.34}$$

where

$$w'_{hh} = w_{ii} + 2w_{ij} + w_{jj}\,, \tag{4.35}$$
$$w'_{hk} = w_{ik} + w_{jk}\,, \tag{4.36}$$
$$w'_{h} = w_i + w_j = w'_{hh} + w'_{hk}\,. \tag{4.37}$$

In the particular case of nodes i and j without self-loops ($w_{ii} = w_{jj} = 0$), the triangular hair can always be reduced to a single hair with a self-loop $w'_{hh} = 2w_{ij}$ (see figure 4.1b).

4.3.2 Reductions for directed networks

Directed networks are considered here in the scope of modularity represented in equation (4.10), although other possibilities have been recently proposed [44]. The treatment of directed networks requires the distinction between the output and input contributions of nodes to modularity:

$$Q = \sum_i q_i^{\text{out}} = \sum_j q_j^{\text{in}}\,, \tag{4.38}$$

where

$$q_{i,r}^{\text{out}} = \frac{1}{2w}\sum_j \left(w_{ij} - \frac{w_i^{\text{out}} w_j^{\text{in}}}{2w}\right)\delta(C_j, r)\,, \qquad q_i^{\text{out}} = q_{i,C_i}^{\text{out}}\,, \tag{4.39}$$

$$q_{j,r}^{\text{in}} = \frac{1}{2w}\sum_i \left(w_{ij} - \frac{w_i^{\text{out}} w_j^{\text{in}}}{2w}\right)\delta(C_i, r)\,, \qquad q_j^{\text{in}} = q_{j,C_j}^{\text{in}}\,. \tag{4.40}$$

The separation of the self-loop term follows the same pattern than for undirected networks:

$$\tilde{q}_{i,r}^{\text{out}} = \frac{1}{2w}\sum_{j(\neq i)} \left(w_{ij} - \frac{w_i^{\text{out}} w_j^{\text{in}}}{2w}\right)\delta(C_j, r)\,, \qquad \tilde{q}_i^{\text{out}} = \tilde{q}_{i,C_i}^{\text{out}}\,, \tag{4.41}$$

$$\tilde{q}_{j,r}^{\text{in}} = \frac{1}{2w}\sum_{i(\neq j)} \left(w_{ij} - \frac{w_i^{\text{out}} w_j^{\text{in}}}{2w}\right)\delta(C_i, r)\,, \qquad \tilde{q}_j^{\text{in}} = \tilde{q}_{j,C_j}^{\text{in}}\,, \tag{4.42}$$

and

$$\tilde{Q} = \sum_i \tilde{q}_i^{\text{out}} = \sum_j \tilde{q}_j^{\text{in}}\,, \tag{4.43}$$

satisfying

$$q_{i,r}^{\text{out}} = \tilde{q}_{i,r}^{\text{out}} + \frac{1}{2w}\left(w_{ii} - \frac{w_i^{\text{out}} w_i^{\text{in}}}{2w}\right)\,, \tag{4.44}$$

$$q_{j,r}^{\text{in}} = \tilde{q}_{j,r}^{\text{in}} + \frac{1}{2w}\left(w_{jj} - \frac{w_j^{\text{out}} w_j^{\text{in}}}{2w}\right)\,, \tag{4.45}$$

and

$$Q = \tilde{Q} + \frac{1}{2w}\sum_i \left(w_{ii} - \frac{w_i^{\text{out}} w_i^{\text{in}}}{2w}\right)\,. \tag{4.46}$$

With these definitions at hand, the change of modularity when node i goes from community r to community s becomes

$$\Delta Q = (\tilde{q}_{i,s}^{\text{out}} + \tilde{q}_{i,s}^{\text{in}}) - (\tilde{q}_{i,r}^{\text{out}} + \tilde{q}_{i,r}^{\text{in}})\,, \tag{4.47}$$

and the change when an isolated node i moves to any community s is

$$\Delta Q = \tilde{q}_{i,s}^{\text{out}} + \tilde{q}_{i,s}^{\text{in}}\,. \tag{4.48}$$

The first difference between directed and undirected networks comes from the fact that now we cannot prove the inexistence of isolated nodes in the partition of optimal modularity. The previous argumentation was based on the use of equation (4.29), which now splits in two relationships:

$$\sum_r \tilde{q}_{i,r}^{\text{out}} = -\frac{1}{2w}\left(w_{ii} - \frac{w_i^{\text{out}} w_i^{\text{in}}}{2w}\right)\,, \tag{4.49}$$

$$\sum_r \tilde{q}_{j,r}^{\text{in}} = -\frac{1}{2w}\left(w_{jj} - \frac{w_j^{\text{out}} w_j^{\text{in}}}{2w}\right)\,. \tag{4.50}$$

The next step is the same:

$$w_{ii} \leq \frac{w_i^{\text{out}} w_i^{\text{in}}}{2w} \quad \Rightarrow \quad \sum_r \tilde{q}_{i,r}^{\text{out}} \geq 0 \quad \Rightarrow \quad \exists s_1 : \tilde{q}_{i,s_1}^{\text{out}} \geq 0 \,, \tag{4.51}$$

$$w_{ii} \leq \frac{w_i^{\text{out}} w_i^{\text{in}}}{2w} \quad \Rightarrow \quad \sum_r \tilde{q}_{i,r}^{\text{in}} \geq 0 \quad \Rightarrow \quad \exists s_2 : \tilde{q}_{i,s_2}^{\text{in}} \geq 0 \,. \tag{4.52}$$

Since communities s_1 and s_2 need not be the same, the change of modularity in equation (4.48) is not guaranteed to be positive, and thus isolated nodes are possible in the partition which maximizes modularity.

Nevertheless, there exist three kinds of nodes for which we can prove that they cannot be isolated in the optimal partition, provided their self-loops are not too large: hairs, *sources* (nodes with only output links) and *sinks* (nodes with only input links).

Directed hairs, i.e. nodes connected only to another node, either through an output, an input, or both links, necessarily have $s_1 = s_2$. Therefore, it is save to group them in the same way as undirected hairs when

$$w_{ii} \leq \frac{w_i^{\text{out}} w_i^{\text{in}}}{2w} \,. \tag{4.53}$$

In particular, if the hair has no self-loop ($w_{ii} = 0$), then this condition is always fulfilled (see figure 4.2a). Whenever the self-loop is present, both output and input links are needed to counterbalance it. The resulting self-loop w'_{kk} of the grouped node has value

$$w'_{kk} = w_{ii} + w_{ik} + w_{ki} \,. \tag{4.54}$$

Sink nodes i are characterized by null output strengths, $w_i^{\text{out}} = 0$, which imply $\tilde{q}_{i,r}^{\text{out}} = 0$ for all communities r. Thus, the change of modularity in equation (4.48) only depends on the value of $\tilde{q}_{i,s}^{\text{in}}$, and expression (4.52) tells us that they can always be grouped with an increase of modularity. The same property applies to sources, which are defined as nodes with null input strengths, $w_i^{\text{in}} = 0$. Note that sources and sinks cannot have self-loops, since this would be in contradiction with their null input and output strengths respectively.

A triangular hair formed by a source node i and a sink node j behaves exactly as the undirected triangular hair, being possible to group them in a single node h with a self-loop (see figure 4.2b), where

$$w'_{hh} = w_{ij} \,, \tag{4.55}$$

$$w'_{hk} = w_{ik} \,, \tag{4.56}$$

$$w'_{kh} = w_{kj} \,. \tag{4.57}$$

4.4 Estimate of the amount of reduction

The above proofs allow us to face the problem of size reduction in complex networks into a firm basis. In particular, this size reduction preserving modularity ensures that the structural mesoscale found by maximizing modularity will be invariant under these transformations. The natural question at this point is:

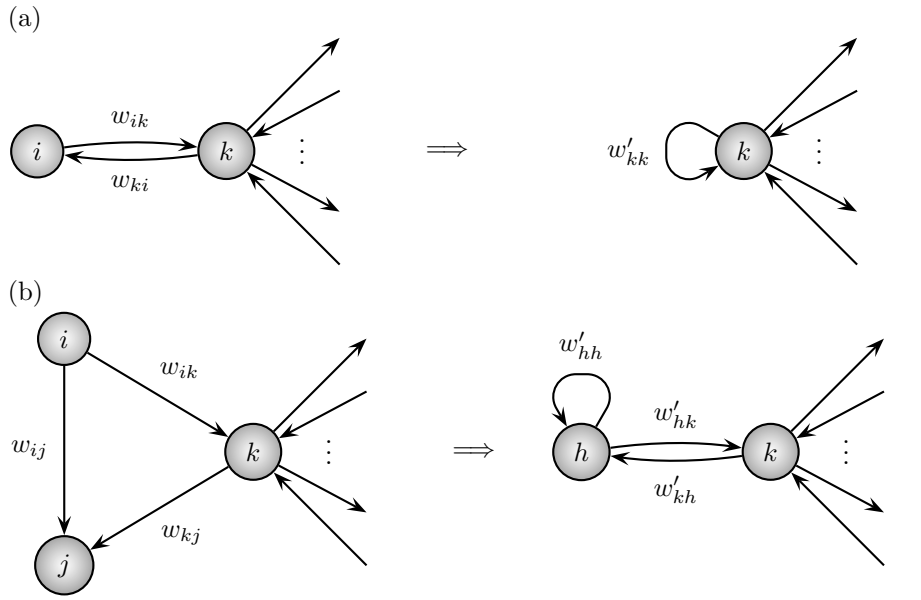


Figure 4.2: Analytic reductions for directed networks: (a) example of a *hair* reduction; (b) example of a *triangular hair* reduction.

what is the percentage in size reduction of networks using the previous rules? To answer this question it is mandatory to have an estimate of the number of hairs and triangular hairs that we might expect in complex networks. In real networks this calculation can be performed by direct enumeration; nevertheless, an estimate can be made in terms of general grounds about the degree distribution $P(k)$.

Here we provide some rough estimates for the most widespread degree distributions in natural and artificial complex networks: scale-free and exponential. For scale-free networks it is usually assumed a $P(k) = \alpha k^{-\gamma}$, with $\gamma \in [2, 3]$ for most of the real scale-free complex networks. The normalization condition provides with the value of α. As a first approximation, neglecting the structural cut-off of the network, we can write

$$\alpha \sum_{k=1}^{\infty} k^{-\gamma} = \alpha \zeta(\gamma) = 1 \,, \tag{4.58}$$

where $\zeta(\gamma)$ is the Dirichlet series representation of the Riemman zeta function. For values of $\gamma \in [2, 3]$ we obtain $\alpha \in [1/\zeta(2), 1/\zeta(3)] \approx [0.61, 0.83]$. This means that, roughly speaking, the number of hairs that corresponds to $P(1)$ is about 83% of nodes in a scale-free network with $\gamma = 3$ and 61% when $\gamma = 2$.

An equivalent estimate can be conducted for exponential degree distributions of type $P(k) = \alpha e^{-\beta k}$, with $\beta > 0$. In this case, normalization implies that

$$\alpha \sum_{k=1}^{\infty} e^{-\beta k} = \alpha \frac{e^{-\beta}}{1 - e^{-\beta}} = 1 \,, \tag{4.59}$$

Table 4.1: Results of the size reduction process for several real networks. For each network we present its number of nodes, before and after reduction, as well as the percentage of size reduction.

Network	N	N_{reduced}	Reduction (%)
Zachary	34	33	2.94
Jazz	198	193	2.53
E-mail	1133	981	13.42
Airports-UU	3618	2763	23.63
Airports-WU	3618	2763	23.63
Airports-WD	3618	2880	20.40
PGP	10680	6277	41.23
AS(2001)	11174	7386	33.90
AS(2006)	22963	15118	34.16

and then $\alpha = e^{\beta} - 1$. The percentage of hairs in this case is $P(1) = 1 - e^{-\beta}$, that for plausible values of $\beta \in [0.5, 1.5]$, provides a reduction between 40% and 77% respectively.

At the light of these estimates, the size reduction process provides with an interesting technique to confront the analysis of community structure in complex networks by maximizing modularity with a substantial advantage in computational cost without sacrificing any information. We have checked our size reduction process in several real networks: the Zachary's karate club network [95]; the jazz musicians network [36]; the e-mail network of the University Rovira i Virgili [41]; the airports network with data about passenger flights operating in the time period from 1 November 2000 to 31 October 2001, compiled by OAG Worldwide (Downers Grove, IL) and analyzed in [42]; the network of users of the PGP algorithm for secure information transactions [15]; and the Internet network at the Autonomous System (AS) level as it was in 2001 and 2006, reconstructed from BGP tables posted by the University of Oregon Route Views Project. The results obtained are reported in table 4.1, where we can observe that the percentage of size reduction is greater than 20% in most cases (this is true in particular for the biggest networks), although the percentage is below our theoretical estimates. This is due to the fact that we have not considered the cut-offs of the degree distributions in real networks.

Particularly illustrative is the analysis of the airports network. We have constructed different networks from the raw data, the unweighted undirected network (airports-UU) previously used in [42], the weighted undirected network (airports-WU) where the weights reflect the number of passengers using the connection in the period of study, and the most realistic case corresponding to the weighted directed network of the airports connections (airports-WD). These networks allowed us to check our reduction technique in all the possible scenarios.

4.5 Summary

In hierarchical clustering, where the results have the form of a nested series of partitions, one needs to use a quality function to know which of the partitions

in the hierarchy is the best one. Several interesting quality functions have been proposed from the field of statistical physics, such as those based on spin models. We have begun this chapter describing one of the most successful functions, the modularity of Newman and Girvan, under a unified framework of quality functions coming out from a particular spin glass model.

The challenge of optimizing the modularity has deserved many efforts from the scientific community in recent years. Provided that the problem is NP-hard, the optimization of modularity cannot be performed by exhaustive search, and only optimization heuristics have proved to be competent in finding suboptimal solutions of the modularity function in feasible computational time. Here we have proposed an exact procedure for size reduction of complex networks preserving the value of modularity, independently on the partition under consideration. First, we have described the property of modularity that allows the size reductions of some complex networks, namely that the nodes forming a community in any optimal partition can be represented by a unique node in a reduced network. Each node in the reduced network summarizes the information necessary for the calculation of the modularity in its self-loop (that accounts for the intra-connectivity of the community) and in its edges (that account for the inter-connectivity with the rest of the network).

Next, we have proved two possible analytic size reductions preserving the modularity of weighted networks (both directed and undirected): hairs, i.e. nodes connected to the network with only one link; and triangular hairs, which are particular structures formed by three nodes. Finally, we have estimated the amount of size reduction (that is, the number of hairs) which one might expect for the most widespread degree distributions in complex networks: scale-free and exponential. We have also performed some experiments of size reduction on several real complex networks.

Chapter 5

Resolution Levels in Complex Networks

The standard approach to determinate the modular structure of complex networks is based on the optimization of the modularity quality function. Initially it was thought that maximizing modularity one always obtains the "best" partition of the network into communities. This partition represents an intermediate topological scale of organization, or *mesoscale*, that in many cases has been shown to coincide with known information about subdivisions in the network [22, 69]. However, recently Fortunato and Barthélemy [32] have proved that the optimization of modularity has an important drawback: the existence of a resolution limit beyond which no separation into smaller groups can be obtained, although these smaller modules might have their own entity. This occurs with any density of links within communities, even at the limit case where all the nodes inside the communities are connected forming cliques. The problem seems to be that modularity, as it has been prescribed, does not have access to these other levels of description which coexist simultaneously and are, in general, a distinctive feature of complex networks. The same limitation has been observed for other quality functions different from modularity [53].

In setion 5.1, we address the issue of community detection in two ways: first, clarifying the conceptual interpretation of the resolution limit, not as a problem but as a feature of quality functions that can help us to understand in deep the structure of networks; and second and most important, we provide with a method that allows the full screening of the topological structure at any resolution level using the original definition of modularity [7]. The main idea is to provide all the nodes with a magnitude, that we call *resistance*, which controls their strengths without affecting the structural topology of the network. Its role is to highlight the contrast between groups whenever they exist. Once the method has been introduced, in section 5.2 we propose a new heuristic algorithm based on Tabu Search, especially well suited for the determination of the mesoscales in networks, and in section 5.3 we validate the method computing the modular structure at multiple resolution levels for several examples of synthetic and real complex networks. Afterwards, in section 5.4 we show some techniques to extract the information hidden in the whole mesoscale found by the algorithm, and in section 5.5 we apply them to the determination of several

scales of organization in the synaptic connectivity of the neuronal system of the *nematode C. elegans* [8]. The general description given all along the application of the method allows its usage on any other complex network. Finally, in section 5.6 we present a brief discussion about the role of the different scales, the comparison of our method with other possible approaches to the mesoscales, and the significance of the mesoscales in contrast with the commonly accepted one-scale of description.

5.1 Networks topology at different scales

5.1.1 Resolution limit and topological scales

Writing the definition of modularity in terms of contribution of modules we have:

$$Q = \sum_{s=1}^{m} \left(\frac{w_{ss}}{w} - \left(\frac{w_s}{2w} \right)^2 \right) , \tag{5.1}$$

where w_{ss} is the internal strength of module s and w_s is the total strength of module s. For unweighted networks w_{ss} reduces to the number of internal links and w_s is the sum of degrees of the nodes in module s.

The solution that we propose takes advantage of the dependence of the resolution limit on the total strength $2w$. Consider the case study consisting on two identical modules with a single link connecting them to the rest of the network and only one link connecting them to each other [32]. The resolution limit states that these modules will not be found, optimizing modularity, if their internal strengths are

$$w_{ss} < \sqrt{\frac{w}{2}} - 1 \, . \tag{5.2}$$

In [32] the authors neglect the contribution -1 in the second side of inequality (5.2), which is acceptable for large values of the total strength.

Our proposal to solve this problem is to modify the total strength $2w$. Let us assume that we increase the strength of every node by a quantity r, then inequality (5.2) will read

$$w_{ss} < \frac{1}{2} \left(\sqrt{2w + Nr} - n_s r - 2 \right) , \tag{5.3}$$

where n_s stands for the number of nodes in module s and N stands for the number of nodes in the network. The result of the prescription (5.3) is that by rescaling the topology by a factor r, the example above can be separated optimizing modularity because the growth of $\sqrt{r}$ is slower than that of r, i.e. at some scale controlled by r both modules will be visible using optimal modularity.

The problem now is how to increase the strength of the nodes without altering the topological characteristics of the original network. We solve this problem by rescaling the topology defining $\mathbf{W}_r$ from the original weighted adjacency matrix $\mathbf{W}$ of the graph, as follows:

$$\mathbf{W}_r = \mathbf{W} + r\mathbf{I} \, , \tag{5.4}$$

where $\mathbf{I}$ is the identity matrix. In terms of graphs, this new matrix represents the original network with self-loops of weight r for every node. Note that the

prescription in equation (5.4) supposes a constant shift (translation) r of the strength of each node.

The commonly analyzed structural characteristics of networks (strength distribution, weighted clustering coefficient, strength correlations of any order, etc.) remain the same in the new network because the translation of strengths does not affect the weights w_{ij} of original links, which are the building blocks of the topology. The shift only affects the property of each node individually and in the same way for all them. The spectra of the original graph is also shifted a quantity r for each eigenvalue, preserving then any property that depends on differences between eigenvalues. The eigenvectors are exactly the same. Finally, the associated Laplacian matrix of the original matrix, $L_{ij} = w_i\delta_{ij} - w_{ij}$, responsible for the behavior of linear dynamical processes on the network [4], remains also unchanged.

The interesting property of the re-scaled topology $\mathbf{W}_r$ is that its characteristic scale in terms of modularity has changed. Then, the topological structure revealed by optimizing modularity for $\mathbf{W}_r$ is that of large groups for small values of r, and smaller groups for large values of r, all of them strictly embedded in the original topology. This fact allows for the screening of the modular structure by analyzing the optimal modular structure of $\mathbf{W}_r$ for different values of r. Note that the rescaling of the topology is simply an elegant way to enhance the total strength of the network, without varying its topological properties. Thus, the rescaling can be used, in principle, to analyze the structure of networks using any quality function at different resolution levels parametrized by r.

5.1.2 Multiple resolution method

The analysis of modules at different resolution levels that we propose, consists into optimize the modularity of the graph $\mathbf{W}_r$ for different values of r. Denoting Q_r the modularity of the network at scale r, the equivalent expression to (5.1) reads:

$$Q_r = \sum_{s=1}^{m} \left(\frac{2w_{ss} + n_s r}{2w + Nr} - \left(\frac{w_s + n_s r}{2w + Nr} \right)^2 \right) . \tag{5.5}$$

The self-link value r for the nodes represents a *resistance*, and stands for the opposition of nodes to become part of any community, in the scope of modularity.

The topological scale determined by maximizing Q at which the detection of modular structure has been attacked so far, corresponds to $r = 0$ (this is the scale at which modularity was originally defined by Newman). For positive values of r, we have access to the substructures below those at $r = 0$, and for negative values of r we have access to the superstructures. The topological scale corresponding to all nodes separated (forming their own communities) is found by maximizing $Q_{r_{\max}}$, where $r_{\max}$ is the smallest positive value of r that satisfies

$$w_{ij} < \frac{(w_i + r)(w_j + r)}{2w + Nr} , \qquad \forall i \neq j . \tag{5.6}$$

And the topological scale corresponding to a unique module formed by the whole network is found by maximizing $Q_{r_{\min}}$, where $r_{\min}$ has a lower bound defined by the asymptote $r_{\text{asymp}} = -2w/N$ (for a detailed analysis see subsections 5.1.3 and 5.1.4). At the asymptote the total strength is zero, thus no meaningful scales

can be found for values of r below it. Note that the average strength can be written as $(2w+Nr)/N = r-r_{\text{asymp}}$. To compare results at different resolution, we adopt the usual formulation in other areas of physics (optics, acoustics, etc.) where scales are prescribed as the logarithm of the ratio between the relevant parameter. Here, the difference between scales, is measured as the logarithm of the ratio between strengths

$$\log\left(\frac{2w+Nr}{2w+Nr'}\right) \equiv \log\left(\frac{r-r_{\text{asymp}}}{r'-r_{\text{asymp}}}\right). \tag{5.7}$$

In this new description, we have that a module is defined at each scale of description r, as the result of the maximization of Q_r. Moreover, modules that exist at a certain level of description may disappear from our observation when changing the scale r while others arise. Note that nothing implies that the substructures to which we will have access at different resolution levels are necessarily hierarchical, indeed in general they will not be hierarchical. Although, in principle, all resolution scales provide some information about the topology, and are important, the detection of partitions that are more persistent than the rest when changing the resolution r is indicative of a tougher modular structure.

5.1.3 Resistance limiting cases for undirected networks

Here we present the mathematical proofs of the physical limiting cases of the resistance for weighted networks, namely the limit of resistance for which all nodes are isolated, and the limit for which all nodes become members of a single group that represents the whole network. In this subsection we deal with undirected networks, whilst next subsection is dedicated to directed networks.

Let $w_{ij} = w_{ji} \geq 0$, $i \neq j$, be the weights of a complex network, where $w_{ij} = 0$ if there is no link between nodes i and j. We suppose that this network is connected; otherwise, each connected component should be analyzed one by one. The addition of a common resistance r to all nodes may be understood as the definition of a new network with weights

$$w'_{ij} = \begin{cases} w_{ij} & \text{if} \quad i \neq j\,, \\ r & \text{if} \quad i = j\,. \end{cases} \tag{5.8}$$

The strengths of this network are

$$w'_i = \sum_j w'_{ij} = w_i + r\,, \tag{5.9}$$

and its total strength is

$$2w' = \sum_i w'_i = 2w + Nr\,. \tag{5.10}$$

Now, the modularity of the new network is calculated as

$$Q_r = \frac{1}{2w'}\sum_i\sum_j \left(w'_{ij} - \frac{w'_i w'_j}{2w'}\right)\delta(C_i, C_j)\,, \tag{5.11}$$

which may also be written as

$$Q_r = \frac{1}{2w'}\sum_i\sum_{j\neq i} \left(w_{ij} - \frac{w'_i w'_j}{2w'}\right)\delta(C_i, C_j) + D_r\,, \tag{5.12}$$

where

$$D_r = \frac{1}{2w'} \sum_i \left(r - \frac{{w'_i}^2}{2w'} \right) . \tag{5.13}$$

Note that D_r does not depend on the community partition.

All nodes isolated

If we have

$$w'_{ij} - \frac{w'_i w'_j}{2w'} < 0 \,, \qquad \forall i \neq j \,, \tag{5.14}$$

then Q_r in equation (5.12) is maximized when $\delta(C_i, C_j) = 0 \,, \forall i \neq j$, i.e. modularity attains its maximum when all nodes are isolated in clusters of just one node. In terms of the resistance they simply become second order inequalities,

$$(2w + Nr)w_{ij} < (w_i + r)(w_j + r) \,, \qquad \forall i \neq j \,, \tag{5.15}$$

which can easily be solved for all pairs of nodes joined by an edge. Thus, $r_{\max}$ is the minimum value of r which satisfies all these inequalities, and for $r > r_{\max}$ all nodes are separated in the optimal community configuration.

All nodes in the same community

Let us analyze the behavior of modularity just to the right of the asymptote $r_{\text{asymp}} = -2w/N$. For convenience, we write the resistance as

$$r = -\frac{2w}{N} + \epsilon \,, \tag{5.16}$$

where $\varepsilon = \epsilon/N$ and ϵ is a small positive constant.

The first term of modularity in equation (5.11) can be split in the following way:

$$\begin{aligned} \sum_i \sum_j \frac{w'_{ij}}{\epsilon} \delta(C_i, C_j) &= \sum_i \sum_j \frac{w'_{ij}}{\epsilon} - \sum_i \sum_{j \neq i} \frac{w_{ij}}{\epsilon} (1 - \delta(C_i, C_j)) \\ &= 1 - \frac{a}{\epsilon} \,, \end{aligned} \tag{5.17}$$

being a the sum of weights of edges connecting different communities. If there are two or more communities, then $a > 0$, otherwise $a = 0$.

The analysis of the second (null case) term in equation (5.11) requires a communities expansion:

$$\begin{aligned} \sum_i \sum_j \frac{w'_i w'_j}{\epsilon^2} \delta(C_i, C_j) &= \sum_c \left(\sum_i \sum_j \frac{w'_i w'_j}{\epsilon^2} \delta(C_i, c) \delta(C_j, c) \right) \\ &= \sum_c \frac{1}{\epsilon^2} \left(\sum_i w'_i \delta(C_i, c) \right) \left(\sum_j w'_j \delta(C_j, c) \right) \\ &= \frac{1}{\epsilon^2} \sum_c \left(\sum_i (w_i + r) \delta(C_i, c) \right)^2 \\ &= \frac{b}{\epsilon^2} \,, \end{aligned} \tag{5.18}$$

where $b > 0$, and $b \sim O(\epsilon^2)$ only if all strengths are equal, on the contrary $b \sim O(1)$.

Therefore,

$$Q_{r_{\text{asymp}}+\varepsilon} = 1 - \frac{a}{\epsilon} - \frac{b}{\epsilon^2}\,, \tag{5.19}$$

which has an asymptotic behavior

$$\lim_{\epsilon\to 0^+} Q_{r_{\text{asymp}}+\varepsilon} = \begin{cases} -\infty & \text{if two or more communities}\,, \\ 0 & \text{if only one community}\,. \end{cases} \tag{5.20}$$

This means that, for values of the resistance just above the asymptote, the optimal communities configuration is that with all nodes together in a single module that corresponds to the whole network.

5.1.4 Resistance limiting cases for directed networks

Let $w_{ij} \geq 0$, $i \neq j$, be the weight of an arc that goes from the i-th to the j-th node, where $w_{ij} = 0$ if there is no link between them. We suppose that this network is connected in the weak sense (weak connected components), i.e. the connected components are found as if the arcs were undirected; otherwise, each connected component should be analyzed one by one.

The natural generalization of modularity to cope with directed networks was introduced in [5], and is expressed as

$$Q_r = \frac{1}{2w}\sum_i\sum_j \left(w_{ij} - \frac{w_i^{\text{out}} w_j^{\text{in}}}{2w}\right)\delta(C_i, C_j)\,, \tag{5.21}$$

where the output and input strengths of the network are

$$w_i^{\text{out}} = \sum_j w_{ij}\,, \tag{5.22}$$

$$w_j^{\text{in}} = \sum_i w_{ij}\,, \tag{5.23}$$

and its total strength is

$$2w = \sum_i w_i^{\text{out}} = \sum_j w_j^{\text{in}} = \sum_{ij} w_{ij}\,. \tag{5.24}$$

The addition of a common resistance r to all nodes may be understood as the definition of a new network with weights

$$w'_{ij} = \begin{cases} w_{ij} & \text{if} \quad i \neq j\,, \\ r & \text{if} \quad i = j\,. \end{cases} \tag{5.25}$$

The strengths of this network are

$$w_i'^{\text{out}} = w_i^{\text{out}} + r\,, \tag{5.26}$$

$$w_j'^{\text{in}} = w_j^{\text{in}} + r\,, \tag{5.27}$$

and its total strength is

$$2w' = \sum_i w_i'^{\text{out}} = \sum_j w_j'^{\text{in}} = 2w + Nr\,. \tag{5.28}$$

Now, the modularity (5.21) of the new network is calculated as

$$Q_r = \frac{1}{2w'} \sum_i \sum_j \left(w'_{ij} - \frac{{w'_i}^{\text{out}} {w'_j}^{\text{in}}}{2w'} \right) \delta(C_i, C_j) \,, \tag{5.29}$$

which may also be written as

$$Q_r = \frac{1}{2w'} \sum_i \sum_{j \neq i} \left(w_{ij} - \frac{{w'_i}^{\text{out}} {w'_j}^{\text{in}}}{2w'} \right) \delta(C_i, C_j) + D_r \,, \tag{5.30}$$

where

$$D_r = \frac{1}{2w'} \sum_i \left(r - \frac{{w'_i}^{\text{out}} {w'_i}^{\text{in}}}{2w'} \right) . \tag{5.31}$$

Note that D_r does not depend on the community partition.

All nodes isolated

If we have

$$\left(w'_{ij} - \frac{{w'_i}^{\text{out}} {w'_j}^{\text{in}}}{2w'} \right) + \left(w'_{ji} - \frac{{w'_j}^{\text{out}} {w'_i}^{\text{in}}}{2w'} \right) < 0 \,, \qquad \forall i < j \,, \tag{5.32}$$

then Q_r in equation (5.30) is maximized when $\delta(C_i, C_j) = 0$, $\forall i \neq j$, i.e. modularity attains its maximum when all nodes are isolated in clusters of just one node. In terms of the resistance they simply become second order inequalities,

$$(2w+Nr)(w_{ij}+w_{ji}) < (w_i^{\text{out}}+r)(w_j^{\text{in}}+r)+(w_j^{\text{out}}+r)(w_i^{\text{in}}+r) \,, \quad \forall i < j \,, \tag{5.33}$$

which can easily be solved for all pairs of nodes joined by an arc. Thus, $r_{\max}$ is the minimum value of r which satisfies all these inequalities, and for $r > r_{\max}$ all nodes are separated in the optimal community configuration.

All nodes in the same community

The analysis of this case follows the same steps as in subsection 5.1.3, yielding also to (5.19):

$$Q_{r_{\text{asymp}}+\varepsilon} = 1 - \frac{a}{\epsilon} - \frac{b}{\epsilon^2} \,. \tag{5.34}$$

The only difference is that now:

$$b = \sum_c \left(\sum_i (w_i^{\text{out}} + r) \delta(C_i, c) \right) \left(\sum_j (w_j^{\text{in}} + r) \delta(C_j, c) \right) . \tag{5.35}$$

Unlike for undirected networks, the value of b is not guaranteed to be positive, and then:

$$\lim_{\epsilon \to 0^+} Q_{r_{\text{asymp}}+\varepsilon} = \begin{cases} -\text{sign}(b)\ \infty & \text{if two or more communities} \,, \\ 0 & \text{if only one community} \,. \end{cases} \tag{5.36}$$

This means that, only if b is positive for all the different community partitions, for values of the resistance just above the asymptote, the optimal communities configuration is that with all nodes together in a single module that corresponds to the whole network. Otherwise, the modularity will raise to $+\infty$ for the maximum modularity configuration, and the single module structure might not be present for any value of the resistance.

5.2 Modularity optimization using Tabu Search

The method to unveil the mesoscales of a complex network consists then in optimizing Q_r, for $r \in [r_{\min}, r_{\max}]$. Different values of r will eventually reveal different optimal partitions that represent intermediate topological scales of the complex network. In order to find the community structure we propose a new method to optimize the modularity based on Tabu Search [37]. The algorithm proceeds as follows: starting from an initial solution (a partition in groups of nodes of the network), *S_Init*, an iterative process that explores the search space begins, stepping from the solution of the current iteration, *S_Iter*, to one of its neighbors, *S_Neig*. The neighborhood is composed by the partitions that are obtained from the current solution by the application of a local operator called *move*. In our case, the *move* operator acts on a node at a time moving it from its current community to another selected at random, or creating a new one. Among the solutions in the neighborhood the best one is chosen to become the new current solution for the next iteration of the algorithm.

In order to escape from local optima, a list of tabu moves is used. This tabu list stores and forbids the most recently accepted moves and it is updated as the algorithm proceeds, so that a move just added to the list is removed from it after a certain number of iterations (*Tabu_Tenure*) have passed. However, tabu moves are allowed when they lead to an improved solution. Once a solution is accepted, the node moved to obtain this solution is inserted into the tabu list, in order to prevent the movement of the same node during the next *Tabu_Tenure* iterations, unless this move leads us to the best solution found until that moment. We have used a logarithmic function on the number of nodes as the number of idle iterations needed to stop the search.

```
function Tabu_Modularity_Optimization
  (Net: Network; {complex network}
  S_Init: Solution) {solution to initiate the search}
  returns (S_Best: Solution) is
const
  Tabu_Tenure: Natural := 5;
var
  Tabu_Moves: Array_Of_Natural; {counters of forbidden moves}
  Max_Idle: Natural; {maximum number of idle iterations}
  Num_Idle: Natural; {number of idle iterations}
  S_Iter: Solution; {solution of the current iteration}
  S_Neig: Solution; {solution in the neighborhood}
  Node_Best: Natural; {node with the best move}
begin
  for Node := 1 to Number_Of_Nodes(Net) do {initialize the tabu moves}
    Tabu_Moves[Node] := 0;
  end for;
  Max_Idle := Maximum_Of_Nonimprovements(Number_Of_Nodes(Net));
  Num_Idle := 0;
  S_Iter := S_Init;
  S_Best := S_Init;
  while Num_Idle < Max_Idle do
    Explore_Neighborhood(Net, S_Iter, S_Best, Tabu_Moves, S_Neig,
```

```
        Node_Best);
    for Node := 1 to Number_Of_Nodes(Net) do {decrease the tabu moves}
      Tabu_Moves[Node] := Maximum(0, Tabu_Moves[Node]-1);
    end for;
    Tabu_Moves[Node_Best] := Tabu_Tenure;
    S_Iter := S_Neig;
    if Modularity(S_Neig) > Modularity(S_Best) then
      S_Best := S_Neig;
      Num_Idle := 0;
    else
      Num_Idle := Num_Idle + 1;
    end if;
  end while;
  return S_Best;
end Tabu_Modularity_Optimization;

procedure Explore_Neighborhood
  (Net: in Network;
  S_Iter: in Solution;
  S_Best: in Solution;
  Tabu_Moves: in out Array_Of_Natural;
  S_Neig: out Solution;
  Node_Best: out Natural) is
var
  S_Move: Solution; {solution from the move of a node}
begin
  Node_Best := 0;
  for Node := 1 to Number_Of_Nodes(Net) do
    S_Move := Solution_From_Move(Net, S_Iter, Node);
    if Modularity(S_Move) > Modularity(S_Best) then
      Tabu_Moves[Node] := 0;
    end if;
    if Tabu_Moves[Node] = 0 and (Node_Best = 0 or else
        Modularity(S_Move) > Modularity(S_Neig)) then
      Node_Best := Node;
      S_Neig := S_Move;
    end if;
  end for;
end Explore_Neighborhood;
```

The main advantage of this algorithm is that it is a mixture of divisive and agglomerative processes, avoiding the drawbacks of each strategy. Moreover, the iterative process can start from any initial partition, which is adequate for the mesoscale determination, since the optimal partitions for nearby values of the resistance are frequently similar. In terms of computational cost, the Tabu Search heuristic is equivalent to other stochastic optimization methods such as Simulated Annealing or Genetic Algorithms.

5.3 Validation of the method

We show the results of our method investigating the modular structure at multiple resolution levels (different scales), for examples of synthetic and real complex networks. A first approach on synthetic networks is illustrative for validation of the procedure when different coexistent topological scales are imposed by construction. We have also analyzed the modular structure of real networks. In general, in real cases, the results are more difficult to assess because nothing from the topology indicates the existence *a priori* of more relevant structure in the network, and only the corroboration *a posteriori* of the structure found with known facts about the (social, biological, etc.) meaning of it can give reliability to any method. In the experiments, we have studied between 100 and 500 values of r inside the interval $(r_{\text{asymp}}, r_{\max}]$ for synthetic networks, and 1000 values of r for real networks. All the experiments have been cross checked using two modularity optimization heuristics: extremal optimization [25], and a new proposal for the optimization of modularity based on Tabu Search, repeating each one 20 times and keeping the partition obtained at the optimal value of Q_r.

In figure 5.1 we have screened the whole range of topological scales for three synthetic networks, representing the number of modules obtained at the optimal partition for Q_r, and the network analyzed highlighting the partition at two representative scales indicated by (I) and (II). Although the networks studied may have more than two relevant scales, we have just drawn two of them chosen among the most representative ones. First we have computed the modular structure in a hierarchical scale-free network with 125 nodes, RB 125, proposed by Ravasz and Barabasi [78]. In figure 5.1a we plot the modular structure found, which shows three different scales that deserve discussion. We observe clearly persistent structures in 5 and 25 communities respectively, that account for the subdivisions more significant in the process, showing two hierarchical levels for the structure. Additionally, the most stable partition in terms of resolution does not correspond to any of the previous ones, but it corresponds to the partition in 26 modules (the same as the one in 25 modules, but isolating the main hub). The partition in 5 modules and the partition in 26 modules are highlighted on the original network. This result is in perfect correspondence with the synchronization patterns produced on this network using coupled oscillators [4].

Another network example used is the H 13-4 network [4], which corresponds to a homogeneous in degree network with two predefined hierarchical levels, being 256 the number of nodes, 13 the number of links of each node with the most internal community (formed by 16 nodes), 4 the number of links with the most external community (four groups of 64 nodes), and 1 more link with any other node at random in the network. In figure 5.1b we represent the network and its corresponding modular structure at different scales. Both hierarchical levels are revealed by the method as they correspond to the original construction of the network: the first hierarchical level consisting in 4 groups of 64 nodes, and the second level consisting in 16 groups of 16 nodes.

Finally, we have used the FB network proposed by Fortunato and Barthélemy [32] to demonstrate the resolution limit of modularity (at $r = 0$). It consists in two cliques of 20 nodes linked with two small cliques of 5 nodes. At $r = 0$ the best partition cannot separate the two small cliques. In figure 5.1c we observe that the partition searched by the authors, formed by the four cliques isolated

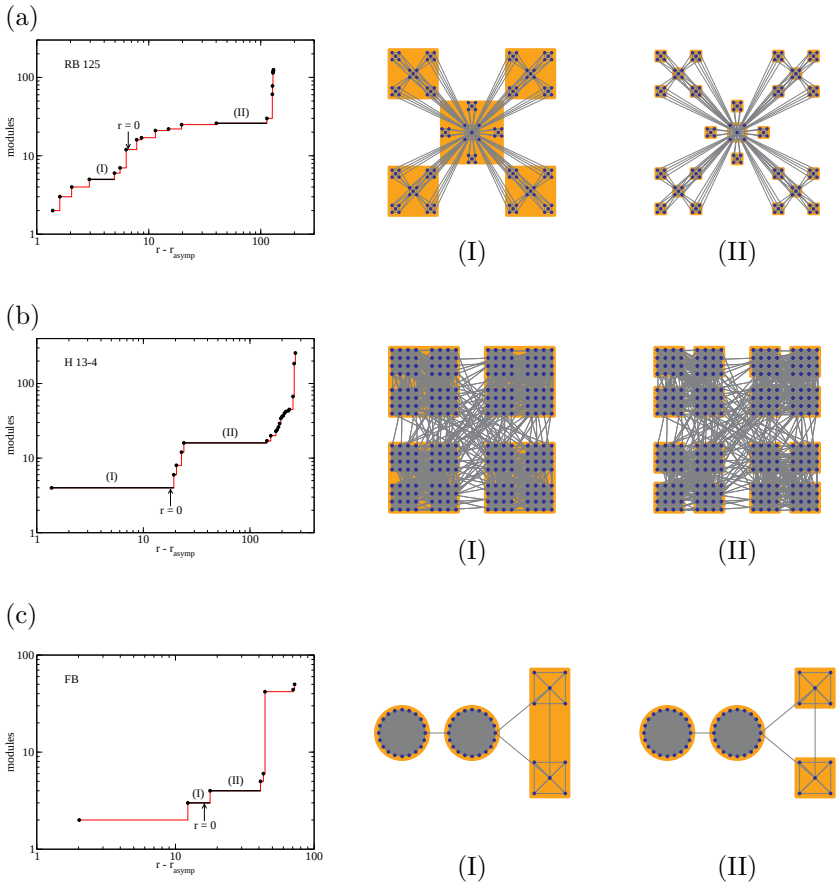


Figure 5.1: Multiple resolution of modular structure in synthetic networks. Left: number of modules obtained at the optimal partition for Q_r, where each point corresponds to a different partition and the arrows indicate the optimal partitions at $r = 0$. Right: networks analyzed, highlighting the partitions at two representative scales indicated by (I) and (II). (a) RB 125 corresponds to the hierarchical scale-free network proposed in [78]. The regions corresponding to 5, 25 and 26 modules are the most representative (stable) in terms of resolution. (b) H 13-4 corresponds to a homogeneous in degree network with two predefined hierarchical levels. Both levels are revealed by the method at different scales. (c) FB corresponds to the network proposed in [32] to demonstrate the resolution limit of modularity (at $r = 0$). This limit is overcome at scale (II) providing with the partition expected.

in their own communities, is obtained by increasing the resolution r, showing that the resolution limit of modularity is overcome by the method in region (II).

We have also studied a couple of social networks for which explicit knowledge about its modular structure is available (see figure 5.2). These particular networks, formed by social acquaintances between individuals, have the main characteristic that after a period of study decomposed in perfectly identifiable parts. The challenge is to find the modular structure of these parts without previous knowledge about the real partition. The optimization of modularity at $r = 0$ fails to provide this information, and no other method has been able to find the real partitioned structure. However, the most representative scales in terms of resolution optimizing Q_r obtained by applying our method correspond exactly to the real splittings.

First, we have investigated the classical social network of the Zachary's karate club [95], accounting for the study over two years of the friendships between 34 members of a karate club at a US university in 1970. The network in question was divided, at the end of the study period, in two groups after a dispute between the club's administrator and the club's instructor, which ultimately resulted in the instructor leaving and starting a new club, taking about half of the original club's members with him. The analysis of this data has been a paradigmatic benchmark to test the accuracy of community detection algorithms. Zachary constructed a weighted network using different social measures (see figure 5.2a), although many times in the literature the network has been considered unweighted for simplicity or tradition, missing important information in the process.

The goal of any community detection algorithm trying to identify modules on this network should be to find the actual split occurred, assigning perfectly the nodes to the two known resulting clubs. The first approach to this goal was given by Girvan and Newman in [35], where they used a divisive method that produced a hierarchical tree representing the whole modular structure. They found that the first network splitting obtained by the method assigned correctly all nodes except node number 3. However, no measure about the quality of the partition was introduced at that time, and then all levels of the hierarchical tree were equivalent, with no way to have a preference for any partition. In [69], the same authors introduced the modularity measure Q and reported that the best structure in the hierarchy, in terms of the value of Q, was a partition in four groups and not two as expected. From this point on, many authors have analyzed this network and have provided the best values of Q obtained. Today it is well accepted that the best partition in terms of modularity of the Zachary's unweighted network is achieved for four groups with a value of $Q = 0.419$. We have applied our method to screen the modular structure of the original weighted network at all resolution scales of r. The results in figure 5.2a show that the most stable level of resolution is precisely the partition resulting in the two groups representing the two clubs, with no mismatch of any individual.

The second network analyzed is the dolphins social network of Lusseau *et al.* [58]. This network was constructed from observations of a community of 62 bottle nose dolphins over a period of seven years from 1994 to 2001. The nodes in the network represent the dolphins, and the ties between nodes represent the associations between dolphin pairs occurring more often than expected by chance. There is evidence [57] that a temporary disappearance of the dolphin denoted SN100, led to the fission of the dolphins community in the two iden-

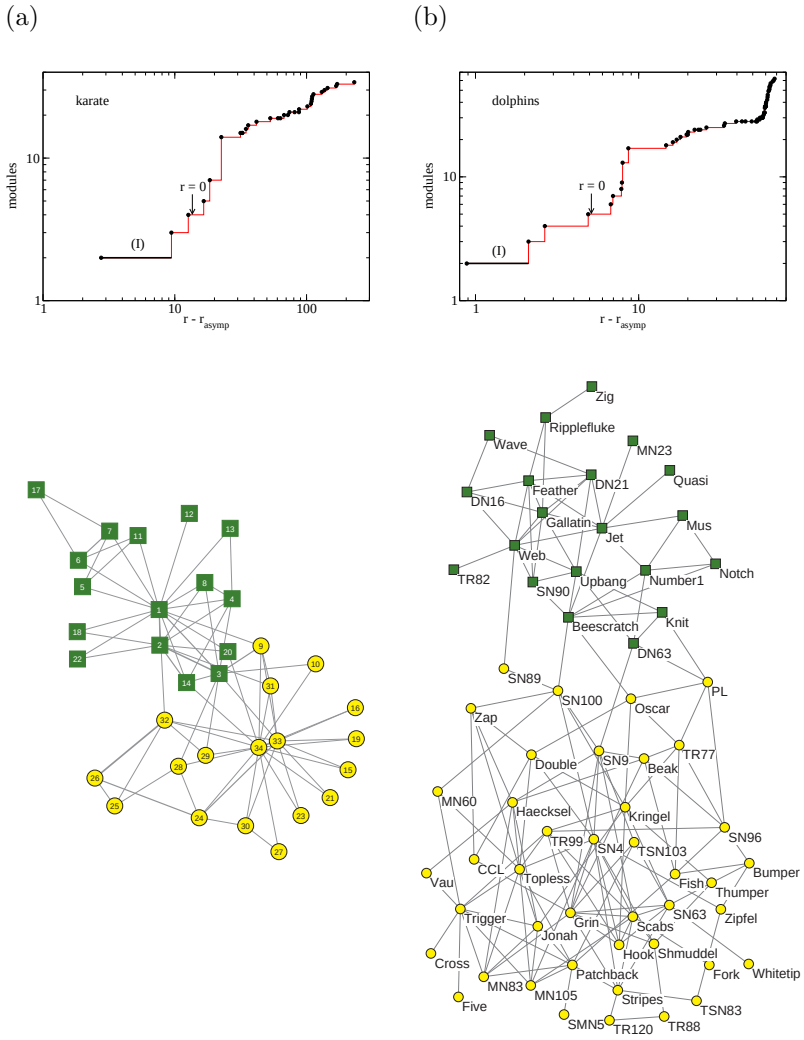


Figure 5.2: Multiple resolution of modular structure in real networks. Top: number of modules in the optimal partitions at different scales; the arrows indicate the best partitions obtained at $r = 0$, which do not correspond to the real partitions. Bottom: representation of the networks and the partitions in the plateaus marked as (I), which correspond both to the most stable scales of description and to the known splittings occurred in the real networks. (a) Zachary's karate club network [95]. (b) Dolphins social network by Lusseau *et al.* [58].

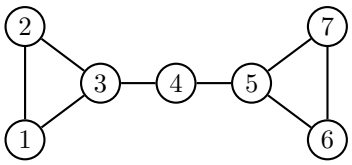


Figure 5.3: Toy network for the calculation of its mesoscales.

tifiable parts shown in figure 5.2b. The optimization of modularity at $r = 0$ does not produce the expected split, but a partition in five communities with $Q = 0.518$. Other approaches as the one exposed in [67] neither success to find the real division. Our method allows to reveal all the modular structure in the whole range of resolution, indicating that the most stable solution in terms of resolution of optimal Q_r corresponds exactly to the two partitions observed in this animal social network.

5.4 Matrices for the analysis of the mesoscales

We have studied the mesoscales in synthetic structured networks and real complex networks [7]. The results show that there are several intermediate scales of description of the complex networks: the topological mesoscales. These scales are revealed by intervals of values of the resistance r, for which the optimal partition does not change.

For a comprehensive representation of the whole mesoscale that allows for the extraction of information, we propose to represent the connectivity matrix after processing it as follows: i) for each pair of nodes we compute the normalized length at values of $r - r_{\text{asymp}}$ in logarithmic scale, for which they belong to the same group, and we represent this normalized length in a color scale; ii) the matrix is reordered from left to right by the size of the connected components with larger lengths at different values of the resistance. The darker colors in the scale represent groups of nodes that are connected at larger lengths.

In the following subsections we give the details of the mesoscales determination for a toy model, in order to clarify all the steps involved.

5.4.1 Mesoscales matrix

Let us consider the undirected graph in figure 5.3, with all weights equal to 1. Its main parameters are: $N = 7$, $2w = 16$, $r_{\text{asymp}} = -2.2857$ and $r_{\max} = 5.2749$. To avoid the infinities at the asymptote and the degeneration of the optimal solution at $r_{\max}$, we study the mesoscales for values of the resistance in the interval $[r_{\text{asymp}} + \varepsilon, r_{\max} + \varepsilon]$, where ε is a small positive constant. In this toy example we have taken $\varepsilon = 10^{-4}$ and a discretization of the resistance range in 1001 equally distributed values. The optimization of modularity has been performed using exhaustive search, appropriate only for very small graphs such as this one. The results are summarized in table 5.1.

When the graph has certain symmetries, a *degeneration* of the optimal community structure may appear, i.e. different optimal partitions of the nodes in communities having the same maximum of modularity. This is the case of the

Table 5.1: Summary of the results obtained during the optimization of modularity for the toy network. We have taken a discretization of the resistance range in 1001 equally distributed values, and the optimization has been performed using exhaustive search.

step	r	$r - r_{\text{asymp}}$	Q_r	size	optimal communities
0	-2.2856	0.0001	0.0000	1	$\{1,2,3,4,5,6,7\}$
1	-2.2781	0.0077	0.0000	1	$\{1,2,3,4,5,6,7\}$
		...		1	$\{1,2,3,4,5,6,7\}$
75	-1.7186	0.5671	0.0000	1	$\{1,2,3,4,5,6,7\}$
76	-1.7110	0.5747	0.0003	2	$\{1,2,3,4\}$ $\{5,6,7\}$
		...		2	$\{1,2,3,4\}$ $\{5,6,7\}$
340	+0.2850	2.5707	0.3808	2	$\{1,2,3,4\}$ $\{5,6,7\}$
341	+0.2926	2.5783	0.3812	3	$\{1,2,3\}$ $\{4\}$ $\{5,6,7\}$
		...		3	$\{1,2,3\}$ $\{4\}$ $\{5,6,7\}$
873	+4.3148	6.6005	0.5221	3	$\{1,2,3\}$ $\{4\}$ $\{5,6,7\}$
874	+4.3224	6.6081	0.5223	5	$\{1,2\}$ $\{3\}$ $\{4\}$ $\{5\}$ $\{6,7\}$
		...		5	$\{1,2\}$ $\{3\}$ $\{4\}$ $\{5\}$ $\{6,7\}$
999	+5.2675	7.5532	0.5541	5	$\{1,2\}$ $\{3\}$ $\{4\}$ $\{5\}$ $\{6,7\}$
1000	+5.2750	7.5607	0.5543	7	$\{1\}$ $\{2\}$ $\{3\}$ $\{4\}$ $\{5\}$ $\{6\}\{7\}$

Table 5.2: Lengths of the mesoscales for the toy network obtained from the range of resistances at which each partition into communities is optimal.

optimal communities	r_{from}	r_{to}	length	% length
$\{1,2,3,4\}$ $\{5,6,7\}$	-1.7110	+0.2926	0.6519	58.25
$\{1,2,3\}$ $\{4\}$ $\{5,6,7\}$	+0.2926	+4.3224	0.4087	36.52
$\{1,2\}$ $\{3\}$ $\{4\}$ $\{5\}$ $\{6,7\}$	+4.3224	+5.2750	0.0585	5.23

configuration in two communities, in which node 4 could have been placed in the other community. In real networks these symmetries do not appear, so they usually do not become a problem.

Any graphical representation of the whole mesoscale should take into account, for every pair of nodes, the proportion of mesoscales at which they belong to the same community. Each mesoscale has a natural *length* (see table 5.2) defined by the range of resistances $[r_{\text{from}}, r_{\text{to}}]$ at which it is optimal:

$$\text{length} = \log(r_{\text{to}} - r_{\text{asymp}}) - \log(r_{\text{from}} - r_{\text{asymp}})\,. \tag{5.37}$$

Thus, the length proportion for a pair of nodes is the sum of the lengths corresponding to mesoscales in which they belong to the same community, normalized by the total length (see figure 5.4a). The graphical representation of this table, which we call *mesoscales matrix*, is shown in figure 5.4b.

This example is quite simple since the mesoscales obtained are hierarchical and then their representation following the hierarchical order is convenient to extract information. However, usually graphs will present non-hierarchical mesoscales, for instance the circular network depicted in figure 5.5 has the mesoscales shown in table 5.3 (with $r_{\text{asymp}} = -2.0$).

(a)

	1	2	3	4	5	6	7
1	1.00	1.00	0.95	0.58	0.00	0.00	0.00
2	1.00	1.00	0.95	0.58	0.00	0.00	0.00
3	0.95	0.95	1.00	0.58	0.00	0.00	0.00
4	0.58	0.58	0.58	1.00	0.00	0.00	0.00
5	0.00	0.00	0.00	0.00	1.00	0.95	0.95
6	0.00	0.00	0.00	0.00	0.95	1.00	1.00
7	0.00	0.00	0.00	0.00	0.95	1.00	1.00

(b)

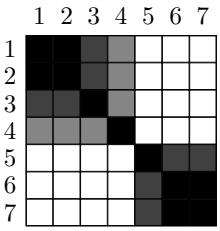


Figure 5.4: Mesoscales table and mesoscales matrix for the toy network. (a) Mesoscales table, formed by the lengths of pairs of nodes in the same community, normalized by the total length. (b) Mesoscales matrix, where the contrast has been adjusted to enhance the visibility of the four different length levels present in the mesoscales table.

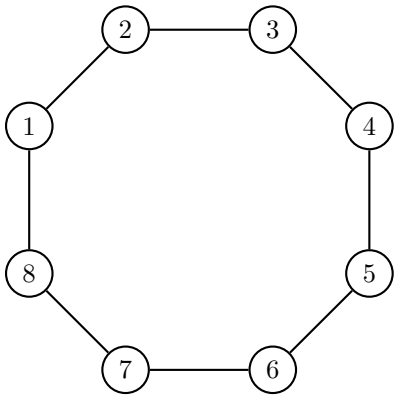


Figure 5.5: Circular network with non-hierarchical mesoscales.

Table 5.3: Non-hierarchical mesoscales for the circular network.

optimal communities	r_{from}	r_{to}	length	% length
$\{1,2,3,4\}$ $\{5,6,7,8\}$	-0.9999	-0.3999	0.2041	22.60
$\{1,2,3\}$ $\{4,5,6\}$ $\{7,8\}$	-0.3999	+0.6721	0.2227	24.66
$\{1,2\}$ $\{3,4\}$ $\{5,6\}$ $\{7,8\}$	+0.6721	+6.0001	0.4762	52.74

Its non-hierarchical structure is clear: nodes 3 and 4 are in the same communities in the first and third partitions, but in different communities in the second one; and nodes 4 and 5 are exactly in the opposite situation. To extract information in this case some processing is necessary, we propose in the next section a possible way to deal with these situations.

5.4.2 Filtered mesoscales matrix

In the previous examples, the mesoscales matrices were simple enough to make all the mesoscales visible at a glance. However, when working with real data, usually the mesoscales matrices turn out to be much more confusing, with disordered nodes and non-transitive relationships of nodes having the same (or similar) lengths. Here we show how to calculate the *filtered mesoscales matrix* from a given mesoscales matrix, as a way to extract the information hidden in it.

Let us suppose that we have obtained the mesoscales table in figure 5.6a, whose mesoscales matrix is shown in figure 5.6b. The filtered mesoscales matrix is obtained by the application of several thresholds to the mesoscales matrix, i.e. the lengths below the threshold are discarded, and the connected components of the graph defined by the remaining lengths are found. Figures 5.6c–f show the results after the application of thresholds 0.25, 0.50, 0.75 and 1.00. The first threshold divides the network in two connected components, which ordered by size are: $\{2, 3, 5, 6, 7\}$, $\{1, 4\}$. This partition gives the reference for the rest of the process. The following connected components, ordered by size within each one of the groups found in the previous threshold, are: $\{2, 5, 7\}$, $\{3, 6\}$, $\{1, 4\}$ for threshold 0.50; $\{2, 7\}$, $\{5\}$, $\{3\}$, $\{6\}$, $\{1, 4\}$ for threshold 0.75; and $\{2\}$, $\{7\}$, $\{5\}$, $\{3\}$, $\{6\}$, $\{1\}$, $\{4\}$ for threshold 1.00. Finally, the filtered mesoscales matrix is built by the composition of these four threshold matrices (see figure 5.6g).

In order to complete this example, we give two more matrices. First, we want to show that by using more threshold cuts in the mesoscales matrix we would obtain a more detailed filtered mesoscales matrix preserving the structures already found because of transitivity. For instance, the result using eight instead of four thresholds is given in figure 5.6h. Second, we would like to assert the difference between the mesoscales matrix and the filtered mesoscales matrix. For this reason we show in figure 5.6i the former using the ordering found by the latter. Clearly, the definition of the filtered mesoscales matrix helps to extract information of the mesoscales imposing transitivity relations in the data found by the mesoscales matrix.

5.4.3 Networks to validate the mesoscales matrices

In order to validate the method we have used synthetic networks where different topological scales coexist. First we have computed the mesoscales in a synthetic hierarchical scale-free complex network, RB 125, which extends up to 125 nodes the model network proposed by Ravasz and Barabasi [78]. This network and its corresponding mesoscales matrix are plotted in figure 5.7 (top). From the mesoscales matrix we observe a clear structure in two hierarchical levels of five and twenty-five communities respectively, that account for the subdivisions more persistent in the process. The significant topological role of the hub connectors is also revealed in this case.

(a)

	1	2	3	4	5	6	7
1	1.0	0.2	0.1	0.8	0.0	0.0	0.1
2	0.2	1.0	0.4	0.0	0.6	0.0	0.8
3	0.1	0.4	1.0	0.2	0.1	0.7	0.1
4	0.8	0.0	0.2	1.0	0.2	0.1	0.0
5	0.0	0.6	0.1	0.2	1.0	0.0	0.1
6	0.0	0.0	0.7	0.1	0.0	1.0	0.3
7	0.1	0.8	0.1	0.0	0.1	0.3	1.0

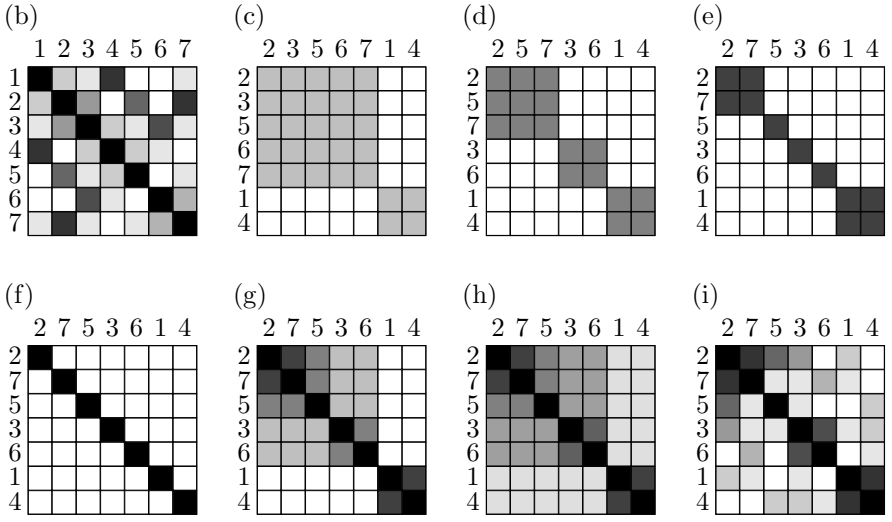


Figure 5.6: Calculation of filtered mesoscales matrices from a mesoscales table. (a) Sample mesoscales table. (b) Corresponding mesoscales matrix. (c) Connected components of the mesoscales matrix at the threshold of 0.25. (d) Threshold of 0.50. (e) Threshold of 0.75. (f) Threshold of 1.00. (g) Filtered mesoscales matrix (4 levels). (h) Filtered mesoscales matrix (8 levels). (i) Mesoscales matrix using the ordering defined by the filtered mesoscales matrix.

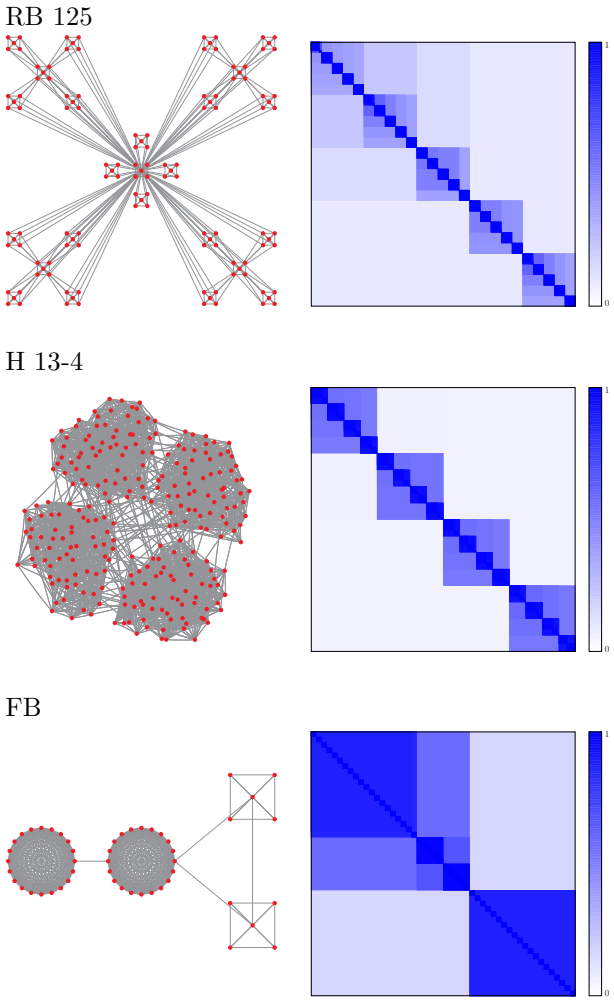


Figure 5.7: Synthetic complex networks and their respective mesoscales matrices. Color levels correspond to the persistence of the structures in r. We plot the networks (left) as well as their mesoscales matrices (right). RB 125 corresponds to an extension of the hierarchical complex network proposed in [78]. H 13-4 corresponds to an homogeneous in degree network with two predefined hierarchical levels. FB corresponds to the network proposed in [32] to demonstrate the resolution limit of modularity.

Another validation example is the H 13-4 network, which corresponds to a homogeneous in degree network with two predefined hierarchical community levels, being: 256 the number of nodes, 13 the number of links of each node with the most internal community (of 16 nodes), 4 the number of links with the most external community (four groups of 64 nodes), and 1 more link with any other node at random in the network [4]. We represent the network and its corresponding mesoscales matrix in figure 5.7 (center). Again the method reveals the hierarchy prescribed with a difference in contrast between the first hierarchical level (groups of 16 nodes) and the second (groups of 64 nodes), as it corresponds to the original construction of the network.

Finally, we have used the FB network proposed in [32] to demonstrate the resolution limit of modularity (at Newman's scale $r = 0$). It consists in two cliques of 20 nodes linked with two small cliques of 5 nodes. At $r = 0$ the best partition cannot separate the two small cliques. In the mesoscales matrix of the FB network drawn in figure 5.7 (down), we observe that the most persistent partition is precisely the one formed by the 4 cliques isolated in their own communities, showing that the resolution limit of modularity can be overcome by the method.

5.5 Analysis of the *C. elegans* neuronal network

Here we develop the analysis of the *C. elegans* neuronal network, from the details of the parameters used to discover its mesoscales, to the enumeration of the functional and anatomical correlations found between neurons.

We have taken the largest connected component of the directed *C. elegans* neuronal network (297 neurons) and we have discretized the resistance range for the determination of the mesoscales in the following way:

1. The interval from $r_{\text{asymp}} = -29.69$ to $r_{\max} = 10357.99$ has been divided in 1000 non-uniform intervals, in such a way that the last resistance increment is ten times larger than the first one, and the size of the increments grow at a constant rate.

2. The significant Newman's scale at $r = 0$ has been added.

3. The negative values of the resistance have been discarded, since we are interested only in sub-structure beyond the standard Newman's scale.

This amounts to the analysis of the mesoscales at 990 different values of the resistance parameter.

The neuronal network of the *C. elegans* can be represented as a weighted directed adjacency matrix (see figure 5.8). The order of the neurons in the matrix follows that of [87], obtained from experimental data in [93]. The detection of the mesoscales in this neuronal system has been performed according to the method explained in this chapter. The best partition at $r = 0$ corresponding to the original Newman's scale provides with five communities. The representation of the obtained groups is depicted in figure 5.9 (left). This figure does not allow the observation of relevant information because of the original order of the neurons in figure 5.8, however after ordering the neurons in the matrix by their communities, the representation shown in figure 5.9 (right) emerges. These

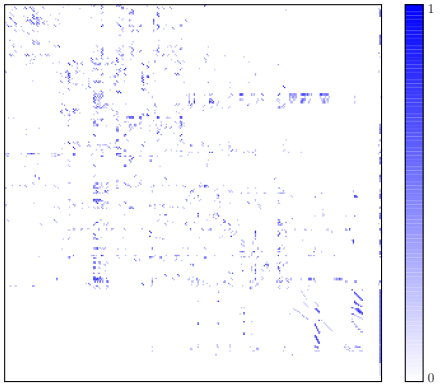


Figure 5.8: Connectivity matrix of the *C. elegans* neuronal network.

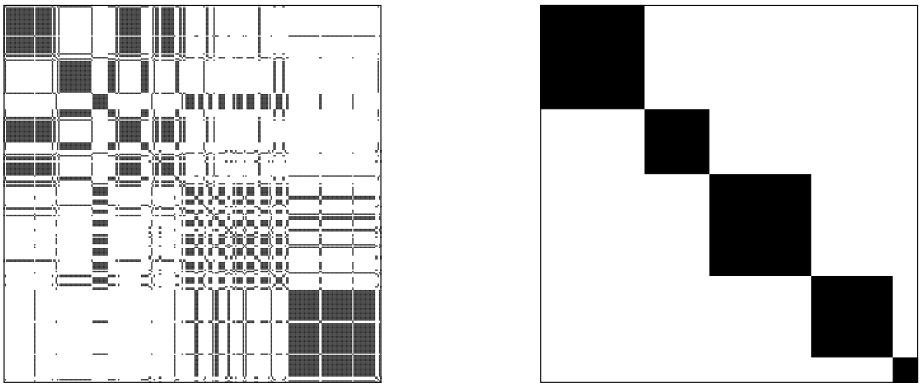


Figure 5.9: Newman's scale of the *C. elegans* neuronal network. Left, original order. Right, reordering by communities.

communities contain neurons whose soma can be correlated with spatial parts of the worm, mainly the head, the body and the tail (see posterior figure 5.13).

The coarse graining at $r = 0$ provides then with a large scale in this system, hence our interest has been focused not in supra-structural but in sub-structural levels. This means that we have analyzed the mesoscales for $r \in [0, r_{\max}]$, obtaining the mesoscales matrix depicted in figure 5.10. We use the partition at $r = 0$ (Newman's scale) as a reference for the substructures found by the method, i.e. Newman's scale corresponds to the threshold equal to 0 in the mesoscales matrix (see figure 5.11).

Any trial of classification by the functional role of neurons in the *C. elegans* is extremely delicate because the multi functional aspects they have. Many neurons participate in different synaptic pathways resulting in different functionalities. This property is also captured by our method showing that at different scales the same neuron can appear in different groups, i.e. the method is not necessarily hierarchical.

In order to extract information from the results obtained, we use the filtered

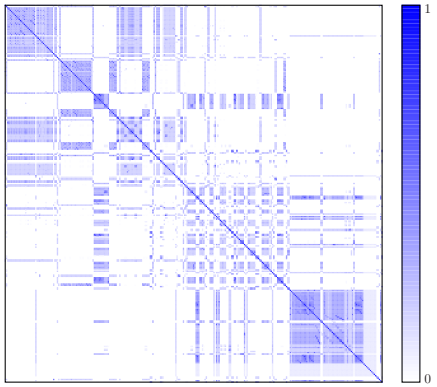


Figure 5.10: Mesoscales matrix for the *C. elegans* neuronal network.

mesoscales matrix as explained in the previous section. By fixing a threshold in the length value, we are able to unravel sub-structural scales that could correspond to groups of neurons involved in different functionalities (see figures 5.11 and 5.12). The most interesting information is that provided at a large value of the threshold, because in this case the substructures found contain small groups of neurons whose activity is most likely associated to a specific action. With this information at hand, and the wide description of each neuron found at the public database of *C. elegans* [93], we propose a tentative classification of some groups of neurons by functionality.

We have studied the filtered mesoscales matrix at a threshold value of 0.6. Fixing our attention at this level of description, we present a tentative functional classification for the groups of five or more neurons (see figure 5.13). We have used the information presented in [26] and [93] for each neuron position and individual functionality, as a guide for the classification of specific actions. Our purpose, after identification of individual functionalities, has been to assign a specific action to the whole group of neurons.

The results of the analysis of the filtered mesoscales matrix for the *C. elegans* neuronal connectivity show that: i) the substructures that prevail at different topological scales are most of them in agreement with the location of the soma of neurons along the body of the worm, and ii) the functionality of the different substructures found by the method are correlated with specific actions of the worm which allows for a tentative classification of functional groups. The classification obtained (see table 5.4) does not pretend to be exact but to provide biologists with a useful information for future research.

5.6 Discussion

5.6.1 Synchronization dynamics

The results show that there exist several intermediate scales of description in complex networks, the topological mesoscale. These scales are revealed by intervals of values of the resistance r, for which the optimal partition does not change (see figure 5.1). The obvious question at this point is: what are these scales

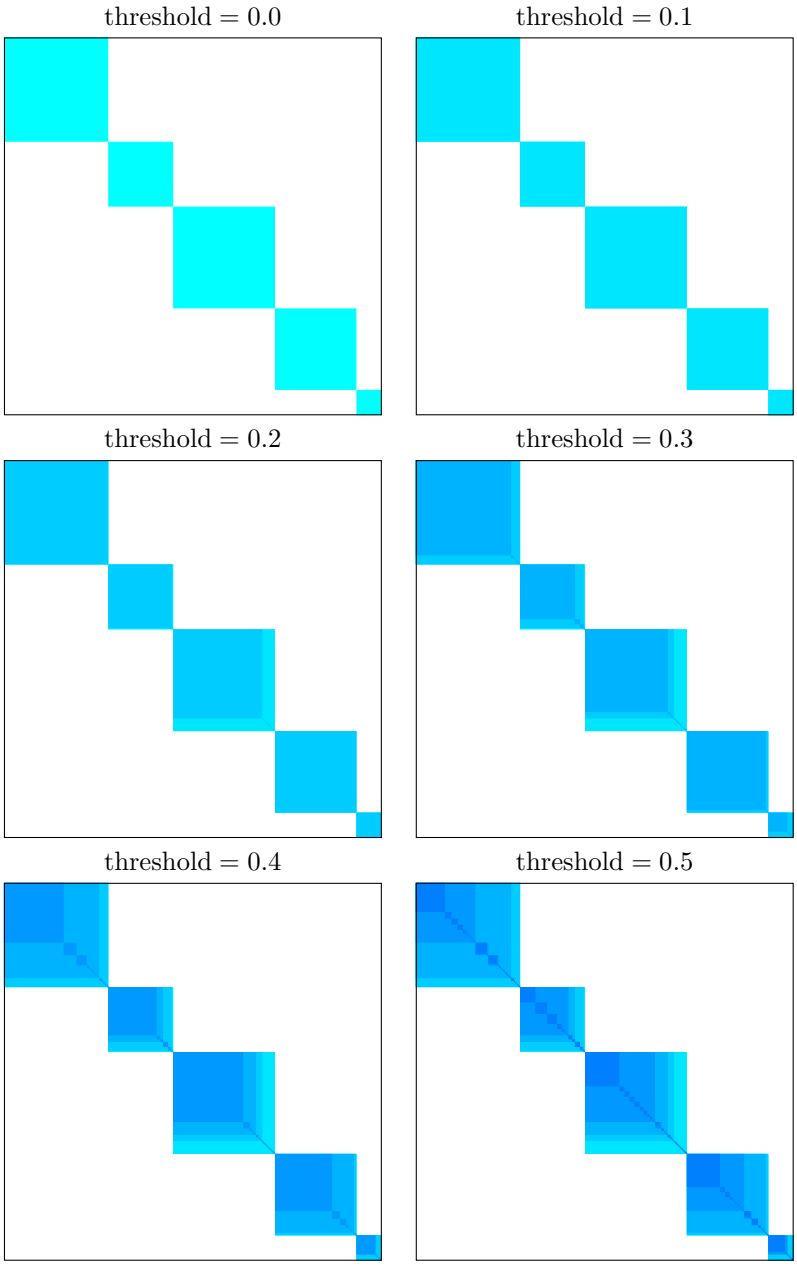


Figure 5.11: Elaboration of the filtered mesoscales matrix for the *C. elegans* neuronal network. Cumulative mesoscales up to thresholds from 0.0 to 0.5.

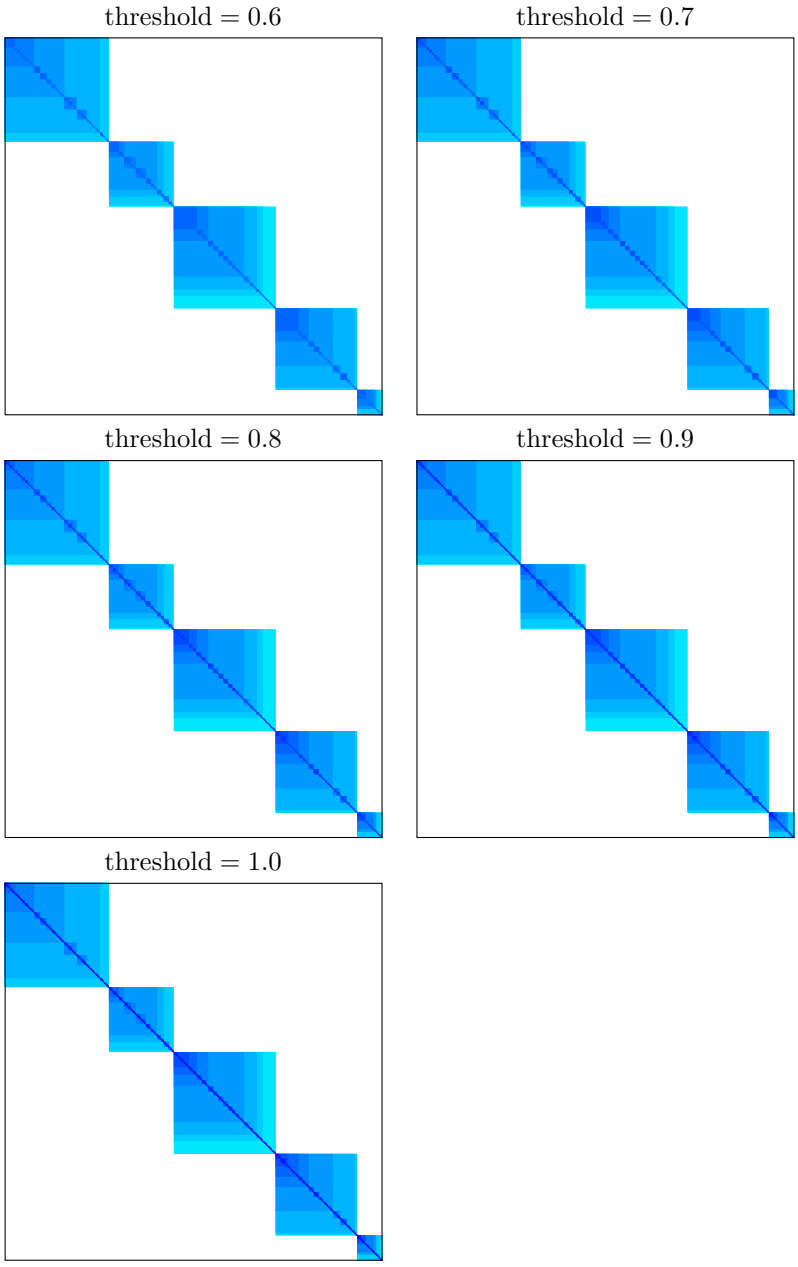
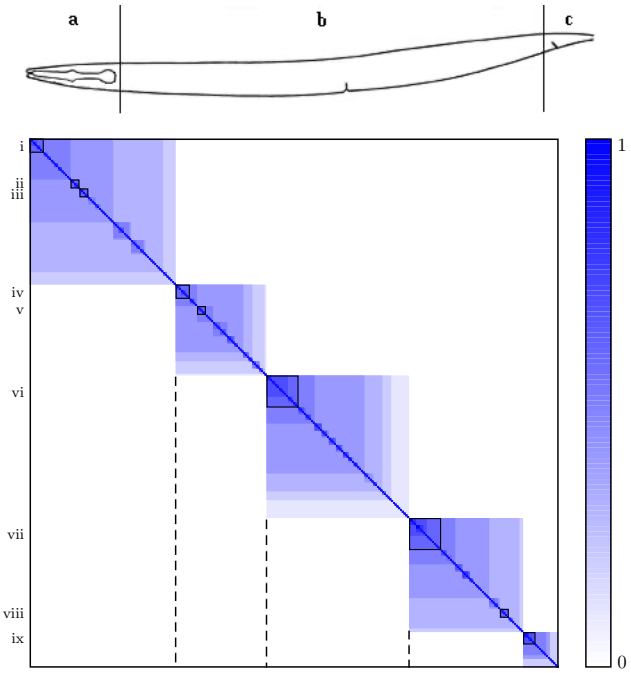


Figure 5.12: Elaboration of the filtered mesoscales matrix for the *C. elegans* neuronal network. Cumulative mesoscales up to thresholds from 0.6 to 1.0.



a	96.4 %	92.0 %	15.2 %	40.6 %	36.8 %
b	1.2 %	0.0 %	82.3 %	32.8 %	26.4 %
c	2.4 %	8.0 %	2.5 %	26.6 %	36.8 %

Figure 5.13: Groups of five or more neurons analyzed from the filtered mesoscales matrix of the *C. elegans* neuronal network at a threshold level equal to 0.6.

Table 5.4: Tentative functionalities for the groups of five or more neurons analyzed from the filtered mesoscales matrix of the *C. elegans* neuronal network at a threshold level equal to 0.6.

i	Neurons:	RIAL, RIAR, RMDDL, RMDR, RMDVR, SMDDR, SMDVL, SMDVR
	Function:	Nose/head orientation movement.
ii	Neurons:	IL1DR, IL1VR, IL2DR, IL2VR, RIPR
	Function:	Head-withdrawal reflex, more related to dorsal relaxation.
iii	Neurons:	IL2L, IL2R, OLQVL, OLQVR, RIH
	Function:	Head-withdrawal reflex, more related to ventral relaxation.
iv	Neurons:	ADLR, AIBR, ASEL, ASHR, AWCL, AWCR, AIAR, AIYL
	Function:	Olfactory and thermo sensation reflex.
v	Neurons:	AIAL, ASGL, ASJL, ASKL, PVQL
	Function:	Chemotaxis to lysine reflex.
vi	Neurons:	AS3, DA2, DA3, DA4, DA5, DB1, DB2, DB3, DB4, DD1, VA3, VB2, VD2, VD3, VD4, VD5, VD6, WM
	Function:	Backward/sinusoidal movement of the worm, more related to touch stimulus.
vii	Neurons:	AVAL, AVAR, AVBL, AVBR, AVDL, AVDR, AVEL, AVER, DA1, FLPL, FLPR, PQR, PVCL, PVCR, PVDL, PVDR, PVPR, RIFR
	Function:	Forward and backward/sinusoidal movement of the worm, more related to search for food in starving case, involve social feeding effect.
viii	Neurons:	AVFL, AVFR, AVHL, AVHR, AVJL
	Function:	Impossible to determine from the experimental data available. There is not any specific function known for any of these neurons.
ix	Neurons:	AVKL, AVKR, DVA, PDEL, PDER, PVM, WN
	Function:	The functionality of this group could be related to a relaxation state similar to a sleep state, with reduced motor activity, decreased sensory threshold, characteristic posture and easy reversibility, basically mediated by PDs neurons.

representative for? The answer of this question is not trivial, and is intrinsically related to the functioning of the complex network as a substrate for different dynamical processes, communication and friendship in social networks, cognitive task in neural networks, or different levels of aggregation of computers in the Internet, for example. Our guess is that a simple dynamical process on top of a complex networks, should somehow reveal the topological mesoscale also in terms of temporal patterns. To check this hypothesis, we have implemented a synchronization dynamics on top of different topologies following [3, 4]. The dynamics corresponds to the non-linear interaction between oscillators connected following the links of the complex networks. Analyzing the temporal metastable patterns emerging in the evolution towards complete synchronization, we corroborate our initial guess.

The temporal mesoscale of the dynamics of synchronization (of phase oscillators) near the synchronization attractor are governed by the solutions of the linear dynamics:

$$\frac{d\theta_i}{dt} = -k \sum_j L_{ij}\theta_j \,, \qquad i = 1, \ldots, N \,, \tag{5.38}$$

where k is a constant, θ_j are the phases of the nodes and L_{ij} the Laplacian matrix of the network.

To identify patterns of synchronization in time, we use [4] a discretization of the matrix $\rho_{ij} = \langle cos(\theta_i - \theta_j) \rangle$ where $\langle \cdot \rangle$ stands for the average over different realizations of the initial conditions. In all cases presented here we have averaged 10^5 realizations, and used a discretization threshold of 0.999. We observe that the intermediate scales that are revealed by the synchronization process are in agreement with those found by the topological method proposed here. The method allows not only to identify the number of communities at different scales but also to determine which nodes form these communities.

We show the corroboration of these claims in a set of synthetic networks, where the modular structure at different scales is imposed by construction. In figure 5.14, we sketch first the topology of a simple model of hierarchical network [78], and the comparison between the specific communities found at different resolution levels, and the synchronization patterns observed in the path towards synchronization. The synthetic network of 25 nodes used combines the scale-free property with a high clustering coefficient, and can be iterated, following the scheme plotted in figure 5.14a, to have many hierarchical levels. The results of the comparison reveal a strong equivalence between both processes, the static resolution method at different scales (different values of the resistance), and the groups of synchronized nodes in time. In figure 5.15 we extend this comparison for three more network structures: H 13-4 corresponding to the homogeneous in degree network described in section 5.3; equivalently the H 15-2 network [4] corresponds to a homogeneous in degree network with two predefined hierarchical levels, being 256 the number of nodes, 15 the number of links of each node with the most internal community, 2 the number of links with the most external community, and 1 more link with any other node at random in the network; the RB 125 network has been used also in section 5.3 and corresponds to the same scheme exposed in figure 5.14a adding a new hierarchical level. The plots here represent, in log-log scale, the number of communities as a function of the translated resistance $r - r_{\text{asymp}}$, and time. The correspondence between

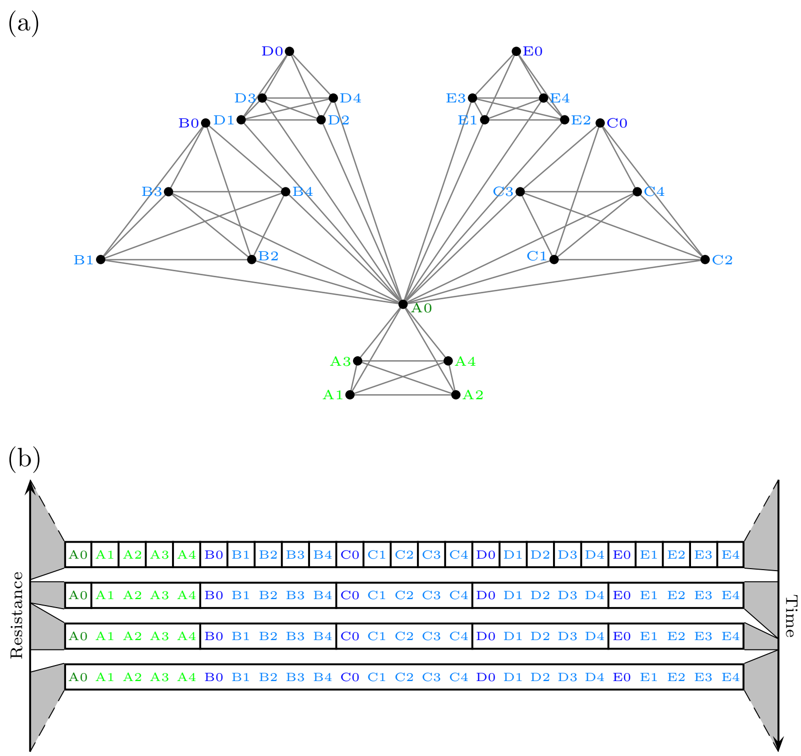


Figure 5.14: Comparison between the communities found at different resolution levels and the groups of synchronized nodes in time. (a) The network structure corresponds to the hierarchical network proposed by Ravasz and Barabasi. (b) Communities found at different scales for the network depicted. The left arrow represents the value of the resistance for which these structures prevail. The right arrow stands for the time intervals for which the same structures are found in a synchronization process. For large positive values of r the network is decomposed in individual nodes, whilst for large negative values of r the whole network forms a single community.

the patterns is highlighted, and again the correspondence is overwhelming.

Obviously, the functioning of real complex networks can rely on dynamical processes very different from the synchronization process exposed here, however it is still instructive to see how a simple nonlinear process reflects the mesoscale of complex networks, or from another point of view, to see how the topology of the networks imposes dynamical (temporal) scales in their functioning.

5.6.2 Comparison with other methods

Some authors have proposed algorithms to extract the hierarchical organization of complex networks by modifying the objective function [79], or by searching local minima of the modularity landscape [80]. These approaches differ conceptually from ours and also in practice: i) the modification of the quality function [79] does not always provide with the correct substructure of networks;

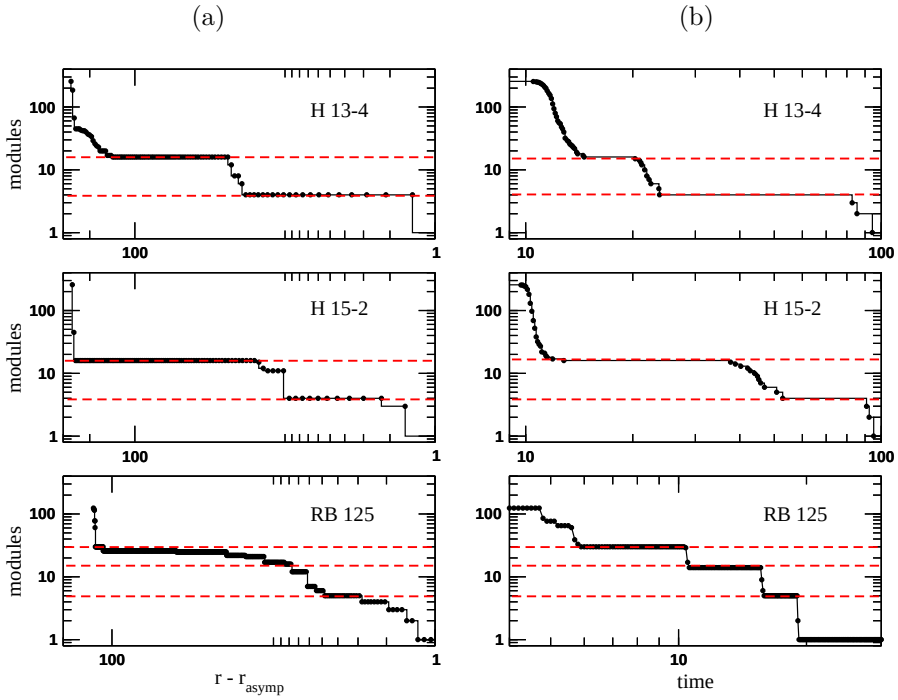


Figure 5.15: Comparison between topological scales and dynamical scales of synchronization. The plots represent, in log-log scale, the number of communities as a function of (a) the translated resistance $r - r_{\text{asymp}}$, and (b) time. Dashed red lines are a guide to the eye to emphasize the correspondence between the plateaus observed. Legends refer to the network structure.

ii) the method based on the screening of local minima of modularity [80] is designed assuming that the structure is hierarchical, which is not the case in many real networks.

The method proposed by Sales-Pardo *et al.* is specifically designed to unravel the hierarchical structure in networks. The comparison with our method is then not possible within our more general scope of any topology. For hierarchical networks, their method will find the hierarchy of scales, as ours also does, however, for non-hierarchical networks, their method can only produce nested communities, in contrast with ours. Conceptually, the difference can be summarized as follows: hierarchies imply multiple scales of description, but the implication does not hold in reverse.

The method proposed by Reichardt and Bornholdt (R&B) [79] was not designed to avoid the resolution limit of modularity but to offer a way to connect modularity with statistical physics. The main idea by the authors is to tune the null model (i.e. change the quality function) and then to obtain other partitions by maximizing the new quality functions. Indeed in [53] the authors interestingly showed that the R&B method has the same resolution problems envisioned in [32] for modularity. The problem is that the R&B method consisting into varying γ (the prefactor that multiplies the null model) is not equivalent to tune the resolution of Newman's modularity. The authors in [53] recognize that if the size distribution of the communities is broad, like in collaboration networks or school friendship networks, there is no single proper value of γ for the optimal resolution. The main difference with our method is that, no matter the size distribution of communities to be broad or not, the rescaling of the topology method that we present finds all the topological structure correctly because it is designed to this end.

To support the above discussion we have built a toy model network with a simple topology but difficult for community detection algorithms because it includes communities of different sizes, some of them sparse and other dense (see figure 5.16). The network model is small enough to have a clear vision of the modules, and to be attacked with computationally costly techniques in reasonable time. While our method succeeds in the process, the R&B method fails. The results of our method and the R&B method varying γ are presented in figure 5.16.

It is worth noticing that the parameter γ in R&B approach does not correspond to any value of r. Only when $\gamma = 1$ and $r = 0$ both definitions become equal, and are exactly Newman's original definition. Rewriting Q_r in equation (5.5) in terms of nodes,

$$Q_r = \frac{1}{2w + Nr} \sum_i \sum_j \left(w_{ij} + r\delta_{ij} - \frac{(w_i + r)(w_j + r)}{2w + Nr} \right) \delta(C_i, C_j) \,, \qquad (5.39)$$

and comparing it to the R&B modularity

$$Q_\gamma^{R\&B} = \frac{1}{2w} \sum_i \sum_j \left(w_{ij} - \gamma \frac{w_i w_j}{2w} \right) \delta(C_i, C_j) \,, \qquad (5.40)$$

for both prescriptions to be equivalent for all partitions, one must show that

$$\frac{(w_i + r)(w_j + r)}{2w + Nr} - r\delta_{ij} = \gamma \frac{w_i w_j}{2w} \,, \qquad \forall i, j \,. \qquad (5.41)$$

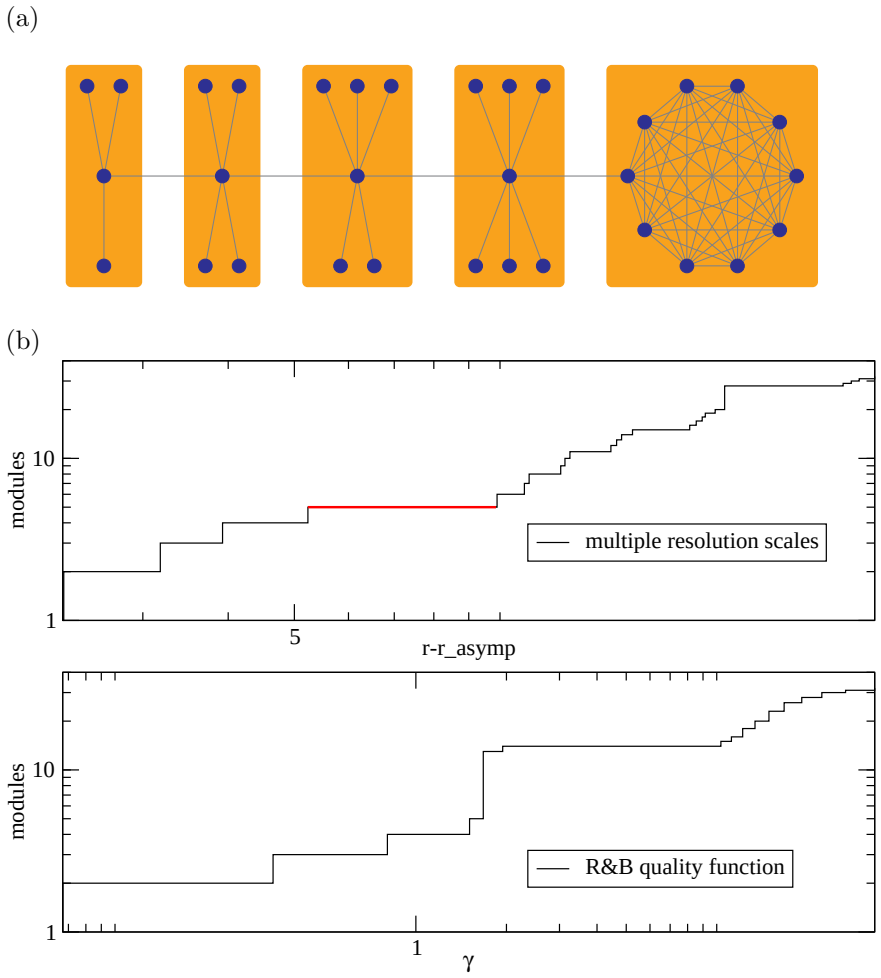


Figure 5.16: Detection of modules in the stars & clique network. (a) Stars & clique network with communities of different sizes and densities of edges. (b) Top: number of modules as a function of the topological scale r using our method; the red line indicates the range of scales for which the natural partition (four stars and one clique) is obtained. Bottom: number of modules as a function of the parameter γ in R&B model; the natural partition (four stars and one clique) is not obtained for any value of γ.

If $i \neq j$, then the relationship between r and γ becomes

$$\gamma = \frac{2w}{2w + Nr} \cdot \frac{w_i + r}{w_i} \cdot \frac{w_j + r}{w_j}, \tag{5.42}$$

which is only fulfilled for $r = 0$ and $\gamma = 1$, the trivial case stated before, because otherwise one would have different values of γ for each pair of nodes i and j. Therefore, there is no transformation of Q_r into $Q_\gamma^{R\&B}$, and they are screening different things.

5.6.3 Which is the "best" scale of description of complex networks?

The question about the determination of the "best" scale of description of a complex network is natural, but ill posed in the current scenario. Throughout the paper we have stated that the more "stable" partitions, in terms of persistence maximizing Q_r when varying the scales with r, are somehow more relevant in the topological description of the mesoscale. Their existence is an observed fact: some partitions are more persistent than others when changing the resolution scale of the topology. We think that this fact is not surprising, as it is not in many physical systems: phenomena that are observed persistently over a wide range of scales vanish at other scales, and others emerge. In general, these more persistent phenomena are usually more important to understand the system. More stable partitions are relevant in the sense that they usually have known meaning, but we cannot state that other partitions not so prevalent are uninformative. All them are embedded in the topology and give their particular information.

Summarizing what we think about the determination of the "best" or "more relevant" scale of description, we can say that the existence of relevant scales of description of a complex network should unavoidably pass through the definition of "relevant". Throughout the paper we have never tried to define "relevant" directly from the results obtained with our method, but *a posteriori*. We use, in the case of synthetic networks and real networks, the information that we have a priori (e.g. knowledge about the hierarchy imposed by construction, or known splits) to determine which scale is more relevant and then to check whether it is found by the method. What we observe is that these relevant scales are usually related to partitions that are significantly persistent (stable) at different scales (variation of r). However, to invert the argument is not straightforward. It is true that one could invent a function that peaks at the scales we see in reality that are known, as for example a function that accounts for the homogeneity of the obtained communities, but this inevitably imposes new conditions to the definition of module. Matching the discussion above let us expose the following: if it exists such a function that indicates the most "relevant" scale of description (and then partition), why not use this function as the objective function to optimize? This argument is strong because it implies that to determine if any scale is more important than others one must optimize a different quality function designed to this end, not modularity.

5.7 Summary

Motivated by the recent finding that the optimization of modularity has a resolution limit related to the characteristic scale imposed by the total strength (sum of weights) of the network, in this chapter we have proposed a multiple resolution procedure that allows the process of optimizing modularity to go deep into the structure. The main idea consists in to re-scale the topology by defining a new network from the original one, providing each node with a self-loop of the same magnitude which we have called resistance. In terms of connectivity, the new network shows the same characteristics that the original network, but allows the search of modules at different topological scales.

To find the community structure, we have proposed a new algorithm to optimize the modularity based on Tabu Search. The results are sets of partitions that screen the full range of structural modules, from individual nodes up to the whole network, in each particular scale. As validation for the method, we have provided examples of the modular substructure and superstructure found in synthetic and real complex networks.

We have also introduced a method to uncover information from the several scales of description found in many real complex systems. A graphical representation of the whole mesoscale is obtained by superposing all the scales found, weighted by the interval in the resistance for which they prevail. The result is a mesoscales matrix whose representation provides with a structural map of the topology of the network. The mesoscales matrices for synthetic networks clearly reveal how nodes form groups at different scales, thanks to the symmetries of the networks analyzed. However, in real complex networks where these symmetries are usually absent, a filtering process is needed to visualize the same information. Therefore, we have designed what we have called the filtered mesoscales matrix, consisting in: i) to fix a mesoscale (a level color) and remove from the mesoscales matrix the elements under this level (lighter colors); ii) to calculate the connected components of the remaining elements (groups); and iii) to arrange the matrix in decreasing size order within the groups obtained at previous levels. The process is iterated starting from the lowest mesoscale to the highest one, accumulating the results of previous stages.

We have applied the complete method to unravel the mesoscales of the neuronal connectivity of the *nematode C. elegans*. The whole nervous system of this worm can be represented as a weighted directed adjacency network. The mesoscales matrix obtained in this case is difficult to analyze because: first, the order of the nodes is not prescribed; and second, the groups obtained at different scales are not necessarily hierarchical. To enhance its visualization we have calculated the corresponding filtered mesoscales matrix.

In the last section of the chapter, we have presented a discussion about the role of the different topological scales beyond its statical definition, revealing their implications in dynamical processes on top of networks. We have also compared our method with other possible approaches to the mesoscales. And finally, we have given a perspective about the significance of the mesoscales in contrast with the commonly accepted one-scale of description.

Chapter 6

General Descriptions of Communities

Even though a lot of work has been done to devise reliable techniques to maximize modularity, very little has been done to analyze the concept of modularity itself and its reliability as a method for community detection. To a large extent, the success of modularity as a quality function to analyze the modular structure of complex networks relies on its intrinsic simplicity. The researcher interested in this analysis is endowed with a non-parametric function to be optimized, the modularity, and the result of the analysis will provide a partition of the network into communities, where each community is a subset of nodes more connected between them than with the rest of the nodes in the network. However, modularity is strongly focused on communities, and for this reason it cannot be used in general to detect groups of nodes revealed by alternative connectivity patterns. The only exception is represented by "anti-communities", i.e. groups of nodes with a few edges inside and many edges connecting different groups. The presence of anti-communities indicates that a network has a multipartite structure. Anti-communities could be detected by modularity minimization [67], although the results are not so good.

Here we propose a general framework to describe groups of nodes, including communities [6]. The contents of the chapter are structured as follows: section 6.1 shows the need to generalize the description of communities given by the standard modularity; then, the mathematical formalism of the generalized modularities is presented in section 6.2; finally, in section 6.3 the framework is tested on synthetic and real networks, and there is a brief discussion about the results obtained.

6.1 Motif-based communities

In general, detecting multipartite structure from first principles requires a definition of the classes that is quite different (in fact, opposite) with respect to standard community definitions. Let us consider bipartite networks, where nodes/actors are connected through other entities, for example collaboration in a work, attendance to an event, etc. In these specific cases, nodes of the same class (e.g. actors) are not directly linked or only share a few edges, and

usually some projection of the network into a subnetwork formed by a single class of nodes is needed for subsequent analysis. For example, in a projection into the actors space, two actors could be connected if they share any team, and the weight of this link could be either one (unweighted projection) or the number of shared teams (weighted projection). However, the analysis of network projections implies the use of partial information and, therefore, some complementary information which could be relevant is missed out. An alternative approach to network projections which allows working with all the available information simultaneously is to generalize the definition of community in order to deal with different classes of nodes. Doing it within a modularity-based framework requires a different formulation of modularity [12, 44].

Bipartite networks are characterized by the fact that any path with even length starting from a node of either class ends in the same class, due to the absence of internal edges in each class. Hence, if the two classes are A and B and we start from a node i_A of class A, the first step leads to one of its neighbors, say i_B, which is in B, the next step to a neighbor j_A of i_B, which is in A, and so on. In this way, paths of even length starting and ending in the same class may reveal bipartite structure, if there are many of them.

On the other hand, in a graph with modular structure, there are many edges inside each module, so one accordingly expects a large number of paths between nodes. In particular, one expects a large number of *cycles*, i.e. closed paths. Small connected subnetworks or *motifs* could be used to define and identify both communities of nodes and more general topological groupings. In fact, the high density of edges within any community determines correlations between nodes that go beyond nearest-neighbors. Here we give different definitions for groups of nodes, including communities, based on the principle that they "contain" more motifs than a null model representing a randomized version of the network at study. The null model of modularity is adopted, i.e. a random network with the same degree sequence of the original network, because modularity lends itself to a simple generalization. Several extensions of modularity are derived, where the building blocks are motifs and not just edges as in the original expression. After that, the new functions are maximized to detect the classes.

The modularity-based framework is used here only as an illustrative example of how motifs can be defined to detect general groups of nodes in networks, but this new framework can be useful to any other method designed to detect substructure in networks. Note that the extended quality functions, which will be introduced, also obey the principle of the resolution limit, which states that modularity will not be able to resolve substructures beyond a certain size limit, just like the original modularity [32]. However this limit is now motif-dependent and then several resolutions of substructures can be achieved by changing the motif.

6.2 Mathematical formulation of motif modularity

The original definition of modularity by Newman and Girvan [69] only deals with unweighted and undirected networks. Later on, Newman generalized it to cope with weighted networks [64]. In this work we start from an extension of

modularity to weighted directed networks [5], which reduces to the previous one for undirected networks, and which is calculated as follows:

$$Q(C) = \frac{1}{2w} \sum_i \sum_j \left(w_{ij} - \frac{w_i^{\text{out}} w_j^{\text{in}}}{2w} \right) \delta(C_i, C_j) \,, \tag{6.1}$$

where w_{ij} is the weight of the connection from the i-th to the j-th node, $w_i^{\text{out}} = \sum_j w_{ij}$ and $w_j^{\text{in}} = \sum_i w_{ij}$ stand for their output and input strengths respectively, $2w = \sum_{ij} w_{ij}$ is the total strength of the network, C_i is the index of the community which node i belongs to, and the Kronecker δ is 1 if nodes i and j are in the same community, and 0 otherwise. For undirected networks, $w_i^{\text{out}} = w_i^{\text{in}} \equiv w_i$, thus recovering the weighted undirected definition of modularity in [64]. The larger the value of modularity, the better the corresponding partition of the network into modules.

In the next subsections we develop the mathematical formulation of a motif modularity which generalizes the standard one given in equation (6.1). First, the most general framework is explained, and then the formalism is applied to several classes of motifs.

6.2.1 General motif modularity

Let $\mathcal{M} = (V_{\mathcal{M}}, E_{\mathcal{M}})$ be a *motif* (connected undirected graph, or weakly connected directed graph), where $V_{\mathcal{M}}$ is the set of M nodes of the motif, and $E_{\mathcal{M}} \subseteq V_{\mathcal{M}} \times V_{\mathcal{M}}$ is the set of its edges.

Let $\{w_{ij} \geq 0 \mid i, j = 1, \ldots, N\}$ be the weights of a (directed or undirected) network of N nodes, where $w_{ij} = 0$ if there is no edge from the i-th to the j-th node, and $w_{ij} \in \{0, 1\}$ if the network is unweighted. The nodes of the motif will be labeled by the indices i_1, i_2, ..., i_M, all of them running between 1 and N.

Given a certain partition C of an unweighted network in communities, the number of motifs fully included within the communities is given by

$$\Psi_{\mathcal{M}}(C) = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_M} \prod_{(a,b) \in E_{\mathcal{M}}} w_{i_a i_b} \, \delta(C_{i_a}, C_{i_b}) \,. \tag{6.2}$$

Degenerated motifs, i.e. those where some nodes are counted more than once, are included in this sum. The formula also holds for weighted networks, which can be inferred from the mapping between weighted networks and unweighted multigraphs [64].

The maximum value of $\Psi_{\mathcal{M}}(C)$ corresponds to the partition in a single community containing all the nodes:

$$\Psi_{\mathcal{M}} = \sum_{i_1=1}^{N} \sum_{i_2=1}^{N} \cdots \sum_{i_M=1}^{N} \prod_{(a,b) \in E_{\mathcal{M}}} w_{i_a i_b} \,. \tag{6.3}$$

For a random network preserving the strength of the nodes, these quantities are respectively

$$\Omega_{\mathcal{M}}(C) = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_M} \prod_{(a,b) \in E_{\mathcal{M}}} w_{i_a}^{\text{out}} w_{i_b}^{\text{in}} \, \delta(C_{i_a}, C_{i_b}) \tag{6.4}$$

and

$$\Omega_{\mathcal{M}} = \sum_{i_1} \sum_{i_2} \cdots \sum_{i_M} \prod_{(a,b)\in E_{\mathcal{M}}} w_{i_a}^{\text{out}} w_{i_b}^{\text{in}} \,. \tag{6.5}$$

Now, by analogy with the standard modularity, we define the *motif modularity* as the fraction of motifs inside the communities minus the fraction in a random network which preserves the strength of the nodes:

$$Q_{\mathcal{M}}(C) = \frac{\Psi_{\mathcal{M}}(C)}{\Psi_{\mathcal{M}}} - \frac{\Omega_{\mathcal{M}}(C)}{\Omega_{\mathcal{M}}} \,. \tag{6.6}$$

The introduction of *null case weights*, n_{ij}, *masked weights*, $w_{ij}(C)$, and *masked null case weights*, $n_{ij}(C)$,

$$\begin{aligned} n_{ij} &= w_i^{\text{out}} w_j^{\text{in}} \,, & (6.7)\\ w_{ij}(C) &= w_{ij}\delta(C_i, C_j) \,, & (6.8)\\ n_{ij}(C) &= n_{ij}\delta(C_i, C_j) \,, & (6.9) \end{aligned}$$

allows the simplification of the previous expressions, in particular motif modularity:

$$Q_{\mathcal{M}}(C) = \frac{\displaystyle\sum_{i_1 i_2 \cdots i_M} \prod_{(a,b)\in E_{\mathcal{M}}} w_{i_a i_b}(C)}{\displaystyle\sum_{i_1 i_2 \cdots i_M} \prod_{(a,b)\in E_{\mathcal{M}}} w_{i_a i_b}} - \frac{\displaystyle\sum_{i_1 i_2 \cdots i_M} \prod_{(a,b)\in E_{\mathcal{M}}} n_{i_a i_b}(C)}{\displaystyle\sum_{i_1 i_2 \cdots i_M} \prod_{(a,b)\in E_{\mathcal{M}}} n_{i_a i_b}} \,. \tag{6.10}$$

Motif modularity may be further generalized by relaxing the condition that all nodes of the motif should be fully inside the modules. This is done just by removing some of the maskings in equation (6.10) as required, and possibly with the addition of some Kronecker δ functions between non-adjacent nodes of the motif. In this way, it is possible to define classes of nodes different from communities, as we shall see in subsection 6.2.3.

6.2.2 Cycle modularity

Among the simplest possible motifs, triangles are the ones which have deserved more attention in the networks literature. For instance, it has been shown that real networks have higher clustering coefficients than expected in random networks [62]. Thus, it would be desirable to be able to find "communities of triangles". Our approach consists in the definition of a *triangle modularity*, $Q_{\triangle}(C)$, based on the triangular motif $E_{\triangle} = \{(1,2),(2,3),(3,1)\}$, which reads:

$$Q_{\triangle}(C) = \frac{\displaystyle\sum_{ijk} w_{ij}(C) w_{jk}(C) w_{ki}(C)}{\displaystyle\sum_{ijk} w_{ij} w_{jk} w_{ki}} - \frac{\displaystyle\sum_{ijk} n_{ij}(C) n_{jk}(C) n_{ki}(C)}{\displaystyle\sum_{ijk} n_{ij} n_{jk} n_{ki}} \,. \tag{6.11}$$

Triangle modularity is trivially generalizable to cycles of length ℓ, making use of the cyclical motif $E_{\mathcal{C}^{(\ell)}} = \{(1,2),(2,3),\ldots,(\ell-1,\ell),(\ell,1)\}$. The number of these motifs within the communities is given by

$$\Psi_{\mathcal{C}^{(\ell)}}(C) = \sum_{i_1 i_2 \cdots i_\ell} w_{i_1 i_2}(C) w_{i_2 i_3}(C) \cdots w_{i_{\ell-1} i_\ell}(C) w_{i_\ell i_1}(C) \,. \tag{6.12}$$

The full formula for the cycle modularity, $Q_{\mathcal{C}^{(\ell)}}(C)$, follows immediately from it.

If the network is directed, then other non-cyclical motifs exist. We skip them, since their derivation is straightforward.

6.2.3 Path modularity

A *path* of length ℓ, $\mathcal{P}^{(\ell)}$, is simply the linear motif defined by the set of edges $E_{\mathcal{P}^{(\ell)}} = \{(1,2),(2,3),\ldots,(\ell,\ell+1)\}$. We remark that cycles are closed paths, but here we shall only consider open paths. The number of paths of length ℓ fully inside the communities is given by

$$\Psi_{\mathcal{P}^{(\ell)}}(C) = \sum_{i_1 i_2 \cdots i_{\ell+1}} w_{i_1 i_2}(C) w_{i_2 i_3}(C) \cdots w_{i_\ell i_{\ell+1}}(C)\,. \tag{6.13}$$

Note that this expression equals the sum of the components of the ℓ-th power of the masked weight matrix.

The path of length $\ell = 1$ corresponds to the simplest motif $E_{\mathcal{P}^{(1)}} = \{(1,2)\}$, which is just a single edge, so its motif modularity (6.10) equals the standard definition of modularity (6.1).

Paths of length 2 are also useful for the analysis of bipartite networks, provided one removes the constraint that all nodes of the path belong to the same module. If one allows that the middle node of a path of length 2 could be any node of the network, whereas the first and third nodes are kept within the same group, the path can be used to discover relationships between nodes of different groups. If a network is bipartite, for instance, there will be many paths of length 2 starting from a class and returning to it from the other class. If only the extremes of the path $\tilde{\mathcal{P}}^{(\ell)}$ are required to be inside the community, their total number is given by

$$\Psi_{\tilde{\mathcal{P}}^{(\ell)}}(C) = \sum_{i_1 i_2 \cdots i_{\ell+1}} w_{i_1 i_2} w_{i_2 i_3} \cdots w_{i_\ell i_{\ell+1}} \delta(C_{i_1}, C_{i_{\ell+1}})\,. \tag{6.14}$$

In this case, the calculation makes use of the ℓ-th power of the weight matrix (instead of the masked weight matrix), and the masking is applied to the sum of their components.

6.3 Examples and tests

When one is faced with the problem of community detection in a particular network, the first thing to do should be to answer the following question: what sort of connectivity patterns or motifs are pertinent in this study? According to the answer, it is straightforward to select one of the possible motif modularities. We present in this section examples of the application of the previous framework to two synthetic networks. Finally, we perform two tests on real networks for which the real partitions observed are known.

The synthetic networks that we have generated for this purpose are the clique & circle network and the star network. In figure 6.1 we show these networks as well as the classes found using different motif modularities. Suppose we want to find node classes by means of triangles. When we optimize the

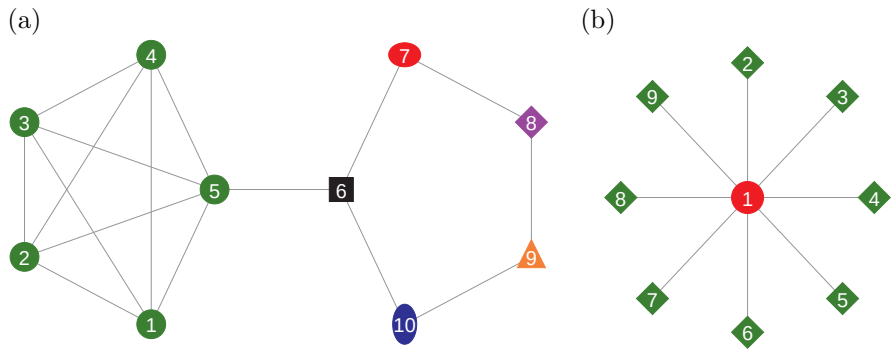


Figure 6.1: Results obtained using motif modularities for two synthetic networks. (a) Clique & circle network, with triangle modularity. (b) Star network, with paths of size 2 modularity with free intermediate node. Members of the same class are depicted using equal symbol and color.

triangle modularity for the clique & circle network, the clique forms a community whereas the nodes of the circle are separated into five singleton communities. This is due to the absence of triangles within the circle. On the contrary, the standard modularity identifies the circle as a community.

The second example, the star network, is a case where the path motifs prove to be useful. This network can be seen as a simple bipartite network with eight actors (the leaf nodes) and just one event (the hub node). In this case, recalling what we have said in the previous section, the path modularity of length 2 with a free intermediate node is the proper motif modularity to use. The results confirm that the star is decomposed in two classes, one for the leaves and another for the hub. The same partition is obtained for any even path length with free intermediate nodes, while for odd path lengths all nodes are joined in a single community. This holds as well if one maximizes the standard modularity; however, the correct partition of the network can be recovered by modularity minimization.

The real networks used for testing are the Zachary Karate Club network [95] and the Southern Women Event Participation network [23, 33]. A description of each network can be found in their respective references. For the mathematical analysis presented here the interesting fact regarding these networks is that we know the real splittings occurred in the Zachary network, as well as the most plausible classification assigned in the literature to the Women Event Participation data, as reported by Freeman [33]. In figure 6.2 we show both networks as well as their respective partitions.

For the Zachary Karate Club network, the nature of the data suggests to try an optimization of path modularities, since the decision of following any of the two leaders during the splitting of the club surely depended on higher order friendship relationships (friends of friends, and so on). When a path modularity of length 1 is considered (i.e. the classical definition of modularity), the best partition obtained splits each one of the two real communities into two subcommunities, yielding a partition in four communities. However, when one looks for a more compact structure of the communities, which can be accomplished by

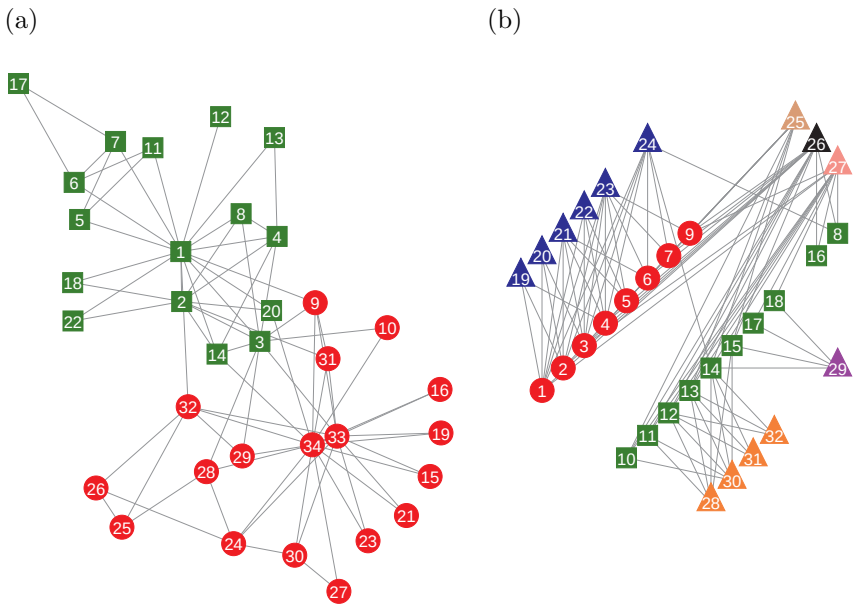


Figure 6.2: Results obtained using motif modularities for two real networks. (a) Zachary Karate Club network, where we depict the real splitting obtained when using several path and cycle modularities. (b) Southern Women Event Participation network, where we depict the results of the analysis of this multipartite network without any projection, simply applying modularity of path free intermediate of length 2. Remarkably the results show clearly the role differentiation of women and events, as well as the splitting of women according to the events participation that has been reported in the literature.

increasing the length of the paths, the optimization of path modularity delivers the real splitting observed, for all path lengths we have used (from 2 to 6). The same result is obtained when the paths are replaced by cycles (lengths from 4 to 9). Triangles give almost the exact partition, but with two exceptions: nodes 10 and 12 become isolated, because they do not belong to any triangle.

The second network tested is a multipartite network. In this case, as well as for the star network, the use of path modularity of length 2 with a free intermediate node is crucial, and it accounts for the role differentiation between women and events. The results not only reveal the two roles of events and women, but also recover their internal split according to their participation in events, a classification made by social scientists [33] (with the same exception of one woman, as in the weighted projection and bipartite methods in [44]). In this case, the minimization of standard modularity is only able to separate women and events, with no further subdivision.

6.4 Summary

In this chapter we have shown that a general classification of groups of nodes in networks is possible if one uses motifs (small connected subnetworks) as elementary units, instead of simple edges. To show that, we have given several definitions for groups of nodes, including communities, based on the principle that they contain more motifs than a null model representing a randomized version of the network at study. Then, we have developed the mathematical formulation for several extensions of modularity where the building blocks are motifs and not edges as in the original expression by Newman and Girvan. First, we have explained the most general framework of motif modularity, and afterwards we have applied this formalism to several classes of motifs. Among the simplest possible motifs, triangles have deserved a lot of attention in the networks literature (for instance, in the definition of the clustering coefficient). Thus, our initial approach has consisted in the definition of triangle modularity, which enables us to find communities of triangles. As a matter of fact, triangles are easily generalizable to cycles, what has immediately led us to the formula for cycle modularity. Finally, we have shown how path modularity is also possible. In fact, paths of length 2 have proved to be very useful for the analysis of bipartite networks.

We have tested these new versions of modularity on synthetic and real networks. Using the type of motif which was pertinent in each case, we have been able to recover the expected connectivity patterns, both when the networks showed modular structure as well as when they presented multipartite structure.

Chapter 7

Conclusions

In this work we have studied several mesoscopic descriptions of complex networks. The main conclusions that can be drawn are the following:

- We have proposed a new variable-group method for agglomerative hierarchical clustering that solves the non-uniqueness problem arising when the classic pair-group method is used. The output of our new algorithm is a uniquely determined type of valued tree, that we have called a multivalued tree, and for which we have devised a new graphical representation called multidendrogram. We can summarize the main advantages of our new proposal in the following points: i) when there are no ties, the variable-group method gives the same result as the pair-group one; ii) the new method always gives a uniquely determined solution; iii) in the multidendrogram representation for the results, one can explicitly observe the occurrence of ties during the agglomerative process and, besides, the height of any fusion interval indicates the degree of heterogeneity inside the corresponding cluster; iv) when ties exist, the variable-group method is computationally more efficient than obtaining all the possible solutions following out the various ties with the pair-group alternative; and v) the new proposal can be also computed in a recursive way using a generalization of Lance and Williams' formula.

- We have studied the correlation-based financial networks and the possibility of filtering such networks into simpler relevant subnetworks using hierarchical asset trees. We have analyzed the space-distortion differences that exist between the output asset trees of distinct hierarchical clustering methods, and we have arranged the results in space-distortion order. At one end of this classification has appeared the single linkage method, suffering from the chaining effect, whilst at the other end of the arrangement we have found the complete linkage and the joint between-within methods, both of them showing clear inner structures corresponding to the branches of their hierarchical trees.

- In order to solve the portfolio selection problem, we have used the hierarchical asset trees to divide the problem into several subproblems according to the clusters arising in the asset trees. We have used the two more promising hierarchical methods for the problem at hand, namely complete

linkage and joint between-within. According to the results obtained, we can conclude that the complete linkage method gives partitions of clusters which are generally better suited in order to divide and solve the portfolio selection problem using our new hierarchical clustering alternative. The time cost of this approach reduces significantly that of the original problem, not only because of the division of the problem into smaller subproblems, but also on account of the possibility to solve the subproblems in parallel since they are disjoint.

- Given that the objective function of the portfolio selection problem resembles so much the energy function of a Hopfield neural network, we have developed a new heuristic method based on the Hopfield neural network to solve the same portfolio selection problem. We have taken the results obtained with our neural network approach and we have compared them to those obtained using three other heuristic methods based on genetic algorithms, tabu search and simulated annealing. All the experimental results lead us to conclude that none of the four heuristic methods has outperformed the others in all types of investment policies. However, we have observed that, when dealing with problem instances demanding proper diversification portfolios with low risk of investment, the neural network model has given better solutions than the other three heuristic methods.

- Provided that the optimization of modularity is an NP-hard problem, it cannot be performed by exhaustive search. Only optimization heuristics have proved to be competent in finding suboptimal solutions of the modularity function in feasible computational time. Here we have proposed an exact procedure for size reduction of complex networks preserving the value of modularity, and we have also estimated the amount of size reductions which one might expect for two of the most widespread degree distributions in complex networks: scale-free and exponential. The systematic use of this size reduction allows to search more exhaustively through the space of partitions, what usually will end in improved values of modularity compared to those obtained without using the size reduction.

- Motivated by the recent finding that the optimization of modularity has a resolution limit related to the characteristic scale imposed by the total strength of the network, we have proposed a multiple resolution procedure that allows the process of optimizing modularity to go deep into the structure. The screening of different scales of descriptions should be useful to get deeper in the understanding of complex networks. The analysis of the results reveals that some topological scales are more persistent (stable) in terms of resolution than others. These stable scales provide with specific information about the main modular aspects of the structure: in the synthetic networks analyzed, they correspond to the predefined structural scales imposed *ad hoc*; and in the real networks, they correspond exactly to previous knowledge about the networks that has not been recovered by any other method studying these networks up to now. With this method, we release optimization of modularity from resolution problems, and give new ideas about the description of complex networks. The existence of several scales of description for complex networks has deep analogies with the common study of complex systems in physics, where different models

have been formulated at different spatial scales to get insight in different aspects of their phenomenology.

- In order to find community structure at different topological scales, we have proposed a new algorithm to optimize the modularity based on tabu search. Its main advantage is that it is a mixture of agglomerative and divisive processes, avoiding the drawbacks of each strategy. Moreover, the iterative process can start from any initial partition, which is adequate for the determination of the mesoscales, since the optimal partitions for nearby values of the resistance are frequently similar.

- We have introduced two techniques for a graphical representation of the several scales of description found in many complex systems. First, the mesoscales matrix which is obtained by superposing all the scales found, weighted by the interval in the resistance for which they prevail. The representation of the mesoscales matrix provides with a structural map of the topology of the network. Second, for instances of real complex networks where symmetries are missing, we have designed a filtering process that produces the filtered mesoscales matrix. This way, without losing any information from the original mesoscales matrix, we achieve a clearer representation of the structural map.

- After calculating the mesoscales matrix and the filtered mesoscales matrix corresponding to the neuronal connectivity of the *nematode C. elegans*, we have unveiled its mesoscales. The results of the analysis show some interesting correlations between the substructures prevailing in the mesoscales and the location of the soma of the neurons or the functionalities in the worm. These results could help biologists to design specific targeted experiments based on the classification of the neurons according to their roles at different topological scales.

- We have given a general classification of groups of nodes in networks developing a unified extension of modularity where the building blocks are general motifs and not simple edges. Using the type of motif which was pertinent in each case, we have tested the new versions of modularity on several networks and we have been able to recover the expected connectivity patterns.

Appendix A

Descripciones Mesoscópicas de Redes Complejas

En las últimas décadas, científicos provenientes de distintos campos (como sociología, biología, física, matemáticas e informática) han estado construyendo la nueva ciencia de las *redes complejas*. Desde Internet y la World Wide Web, hasta las redes de amistades e incluso las redes de transmisiones de enfermedades, la realidad de las redes está en casi todos los ámbitos de la sociedad moderna. Los científicos han observado que muchos sistemas reales están estructurados en forma de redes complejas, esto es, en forma de grafos que representan las conexiones existentes entre sus elementos [11, 85, 87]. Pero la primera pregunta que uno se plantea es: qué es exactamente una red compleja? La respuesta a esta pregunta es sencilla, pues una red no es nada más que un conjunto de elementos (llamados nodos o vértices) y un conjunto de arcos que conectan a los elementos de la red por pares. Algunos ejemplos habituales de redes complejas incluyen: sistemas tecnológicos como Internet [1, 27] y la World Wide Web [2, 31]; sistemas biológicos como redes de interacción de genes o proteínas [40, 46, 73]; una gran variedad de redes sociales [35, 36, 70]; mercados financieros [60]; e infraestructuras de transporte como ferrocarriles y rutas aéreas [42].

La nueva ciencia de las redes complejas es importante por varios motivos. Uno de ellos es que, al fijarse en las propiedades de redes reales, estudia la estructura de las redes tal y como aparecen de manera natural en el mundo real. Las redes sociales y las redes biológicas son ejemplos de este tipo que surgen de manera natural, al igual que también lo son las redes de información como las redes de citas y la World Wide Web. Además, los modelos teóricos pertinentes también son esenciales para poder comprender correctamente el significado de cualquier hallazgo empírico. Por lo tanto, la observación empírica y el modelado teórico se estimulan continuamente el uno al otro.

Otra característica distintiva de la ciencia sobre redes complejas es que intenta establecer la relación existente entre las propiedades estructurales de un sistema y su comportamiento. Las redes compleja no sólo tienen propiedades topológicas, sino que también presentan propiedades dinámicas. Bajo este punto de vista, los vértices de una red representan entidades discretas y dinámicas, con sus propias reglas de comportamiento, mientras que los arcos representan conexiones entre estas entidades.

La descripción *macroscópica* de redes complejas en términos de propiedades estadísticas se ha desarrollado ampliamente, a la búsqueda de su clasificación universal. Entre estas propiedades encontramos el llamado *efecto de mundo pequeño*, que dice que la distancia promedio entre los nodos de una red es pequeña, y habitualmente escala de manera logarítmica con el número total de nodos en la red. Otra propiedad macroscópica presente en muchas redes complejas es la característica distribución de grado en forma de ley de potencias, lo que significa que típicamente hay muchos nodos con grados bajos y unos pocos nodos con grados elevados, siguiendo a menudo dicha distribución una forma exponencial o de ley de potencias. Una tercera propiedad que muchas redes tienen en común es la transitividad (o elevado coeficiente de clustering), que consiste en que dos nodos vecinos de un mismo tercer nodo tengan una mayor probabilidad de ser ellos mismos vecinos entre sí.

Cuando las redes complejas son analizadas de manera local, emergen algunas características que permanecían parcialmente ocultas en la descripción estadística. La más relevante quizás sea el descubrimiento de la *estructura de comunidades* en muchas de ellas [35], según la cual los nodos de una red se reúnen dentro de grupos de nodos conectados entre sí de una manera fuerte o densa, con conexiones más esparsas o débiles entre grupos distintos. Por ejemplo, considerando el caso de las redes sociales (redes de amistades u otras relaciones entre individuos), se observa habitualmente que tales redes contienen comunidades en su interior: subconjuntos de nodos dentro de los cuales las relaciones nodo a nodo son bastante densas, mientras que las relaciones entre subconjuntos distintos no son tan densas.

El estudio de la estructura de comunidades en redes complejas ha recibido mucha atención en los últimos años [22, 65]. En este trabajo nos centramos exactamente en el análisis de varias *descripciones mesoscópicas* de redes complejas. Es un tema interesante porque puede ser muy valioso identificando estructuras a un nivel de descripción mesoscópico, las cuales podrían revelar información acerca de la funcionalidad de grupos de nodos [40, 43]: las comunidades en una red social pueden representar agrupaciones sociales reales, quizás según intereses comunes o formación; las comunidades en una red de citas pueden representar artículos sobre un mismo tema relacionados; las comunidades en una red metabólica pueden representar ciclos u otras agrupaciones funcionales; las comunidades en la World Wide Web pueden representar páginas sobre temas relacionados. Ser capaces de identificar estas comunidades puede ayudarnos a comprender mejor las redes y explotarlas de manera más eficiente.

Clustering jerárquico

Los métodos tradicionales para detectar estructura de comunidades en redes están tomados del análisis de las redes sociales y se reúnen bajo el nombre de *clustering* [20, 38, 84]. Los métodos de clustering agrupan individuos en grupos de individuos o *clusters*, de manera que los individuos dentro de un cluster están cercanos los unos de los otros. Se clasifican en dos grandes grupos, aglomerativos y divisivos, dependiendo de si consisten en reunir o separar comunidades. Los métodos aglomerativos se han usado más habitualmente porque son más eficientes en tiempo de cálculo que la alternativa divisiva. En los métodos aglomerativos se empieza definiendo una medida de similitud y calculando sus valores

entre todos los pares de nodos de la red. A continuación se comienza un proceso iterativo partiendo de tantos clusters individuales como individuos existan, y a cada nuevo paso se fusionan los dos clusters más similares hasta que todos los individuos se hallan en un mismo cluster. El clustering jerárquico no proporciona una única partición de la red en comunidades, sino que los cortes a través de diferentes niveles del árbol nos proporcionan todo un conjunto mesoscópico de particiones anidadas. Hemos comenzado el capítulo 2 describiendo el algoritmo clásico de clustering jerárquico de grupo par, y recordando la definición de árbol valuado, que es el resultado del algoritmo aglomerativo de grupo par y que se representa gráficamente mediante un *dendrograma*.

Entre los distintos tipos de métodos aglomerativos encontramos los de enlazado sencillo, enlazado completo, promedio no pesado, promedio pesado, etc., los cuales difieren entre sí por la manera como llevan a cabo el proceso iterativo que va de los clusters individuales al cluster final. Excepto en el caso del enlazado sencillo, todos los otros métodos aglomerativos de grupo par sufren un problema de no unicidad cuando dos o más valores de similitud entre clusters distintos coinciden durante el proceso de agrupamiento. Esto es la causa de que se generen distintas clasificaciones jerárquicas a partir de un mismo conjunto de datos que representan proximidades y en el cual aparecen empates. En tales casos, seleccionar una única clasificación puede provocar resultados no deseados. Este problema se ha venido tratando tradicionalmente con distintos criterios, la mayoría de los cuales consisten en la selección de una jerarquía resultante entre varias posibles. Nosotros hemos propuesto un algoritmo de grupo variable que consiste en agrupar más de dos clusters simultáneamente cuando aparezcan empates, resolviendo así el problema de la no unicidad en el clustering jerárquico aglomerativo. La salida de este algoritmo es un tipo de árbol valuado y unívocamente determinado, al cual hemos denominado árbol multivaluado, y para el cual hemos ideado una representación gráfica nueva llamada *multidendrograma*. Además, hemos ilustrado la utilidad de nuestra propuesta con algunos resultados correspondientes a datos de un ejemplo real formado por los valores de similitud entre veintitrés suelos distintos (ver figura A.1). Este ejemplo había sido previamente analizado por otros autores, quienes habían detectado la existencia de un valor duplicado en la matriz de entrada que originaba dos posibles jerarquías distintas como resultado de usar el enfoque de grupo par. El uso de nuestra alternativa de grupo variable conduce a un único resultado que coincide con la clasificación conocida para dichos datos de suelos.

A continuación hemos recordado las diferentes definiciones de distancia entre clusters usadas por los métodos jerárquicos aglomerativos más habituales (enlazado sencillo, enlazado completo, promedio no pesado, promedio pesado, centroide no pesado, centroide pesado, e inter-intra), y las hemos generalizado con el propósito de poder utilizar el algoritmo de grupo variable. La utilización de cualquiera de estos métodos aglomerativos especifica implícitamente modelos para los datos y puede proporcionar resultados confusos acerca de la estructura de clases presente en los datos estudiados. Hemos revisado la idea de distorsión del espacio en el clustering jerárquico, según la cual las estrategias pueden ser espacio-conservadoras, espacio-contractoras o espacio-dilatadoras, y hemos redefinidos estos conceptos en términos del nuevo enfoque de grupo variable.

Finalmente, hemos generalizado la fórmula de Lance y Williams, la cual permite obtener clasificaciones jerárquicas aglomerativas de manera recursiva, que permite implementar el algoritmo de manera recursiva. En su formulación,

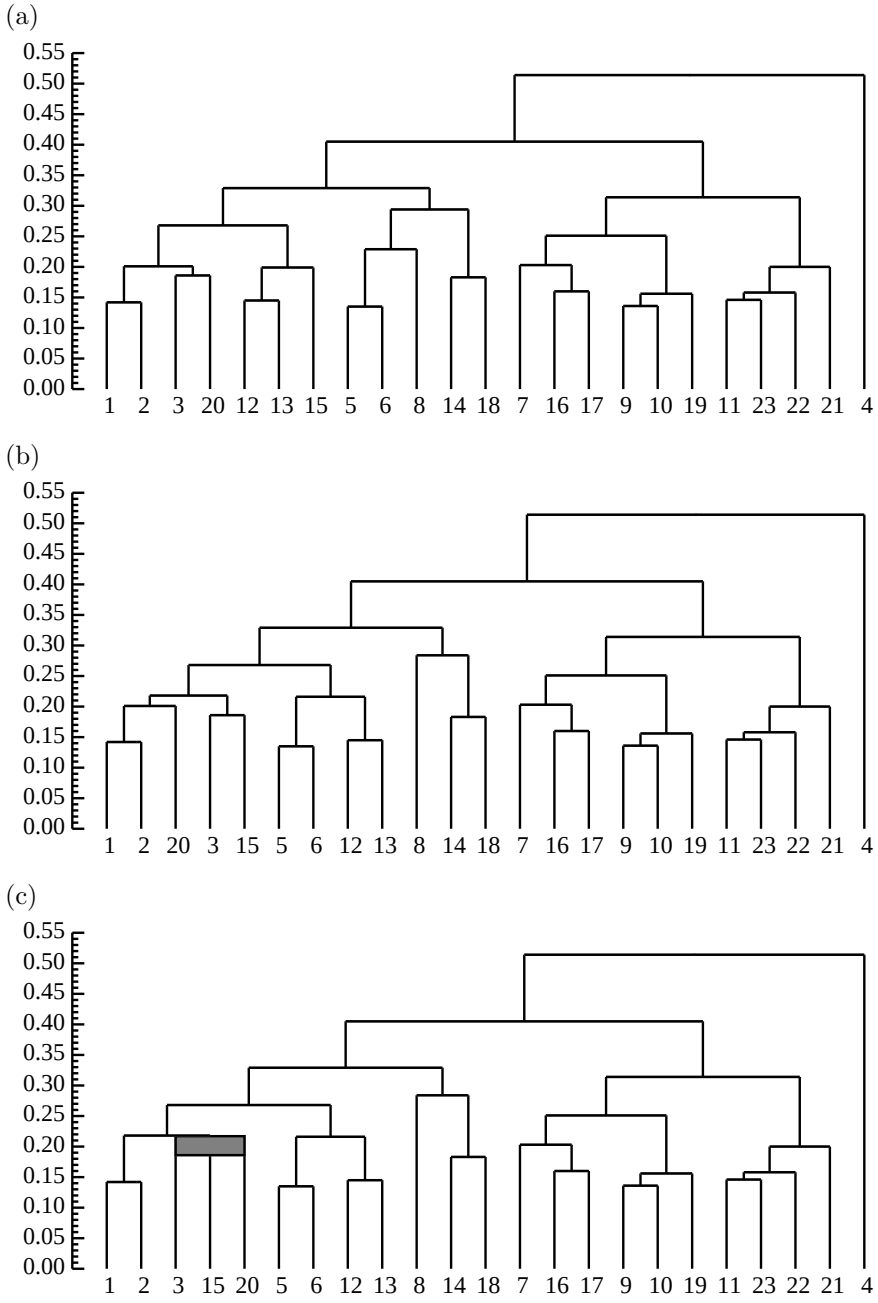


Figure A.1: Dendrogramas y multidendrograma de enlazado completo para los datos de suelos. Teniendo en cuenta el grupo de suelos de tierras marrones formado por los suelos 1, 2, 6, 12 y 13, se puede observar que el dendrograma en (a) es peor que el de (b) porque el primero reúne dichos suelos en una etapa posterior del proceso de agrupación. El correspondiente multidendrograma (c) para estos mismos datos nos proporciona la clasificación correcta de los suelos.

Table A.1: Parámetros para la fórmula de grupo variable.

Método	α_{ij}	$\beta_{ii'}$ $(\beta_{jj'})$	γ_{ij}	δ
Enlazado sencillo	$\frac{1}{\|I\|\|J\|}$	0	$\frac{1}{\|I\|\|J\|}$	0
Enlazado completo	$\frac{1}{\|I\|\|J\|}$	0	$\frac{1}{\|I\|\|J\|}$	1
Promedio no pesado	$\frac{\|X_i\|\|X_j\|}{\|X_I\|\|X_J\|}$	0	0	–
Promedio pesado	$\frac{1}{\|I\|\|J\|}$	0	0	–
Centroide no pesado	$\frac{\|X_i\|\|X_j\|}{\|X_I\|\|X_J\|}$	$-\frac{\|X_i\|\|X_{i'}\|}{\|X_I\|^2}$	0	–
Centroide pesado	$\frac{1}{\|I\|\|J\|}$	$-\frac{1}{\|I\|^2}$	0	–
Inter-intra	$\frac{\|X_i\|+\|X_j\|}{\|X_I\|+\|X_J\|}$	$-\frac{\|X_J\|}{\|X_I\|}\frac{\|X_i\|+\|X_{i'}\|}{\|X_I\|+\|X_J\|}$	0	–

suponemos que deseamos agrupar simultáneamente dos familias de clusters, respectivamente indexadas por $I = \{i_1, i_2, \ldots, i_p\}$ y $J = \{j_1, j_2, \ldots, j_q\}$, en dos superclusters $X_I = \bigcup_{i\in I} X_i$ y $X_J = \bigcup_{j\in J} X_j$. Definimos la distancia entre ellos, $D(X_I, X_J)$, en términos de las distancias entre los respectivos clusters componentes, $D(X_i, X_j)$, como:

$$\begin{aligned} D(X_I, X_J) = & \sum_{i\in I}\sum_{j\in J} \alpha_{ij} D(X_i, X_j) \\ & + \sum_{i\in I}\sum_{\substack{i'\in I \\ i'>i}} \beta_{ii'} D(X_i, X_{i'}) + \sum_{j\in J}\sum_{\substack{j'\in J \\ j'>j}} \beta_{jj'} D(X_j, X_{j'}) \\ & + \delta \sum_{i\in I}\sum_{j\in J} \gamma_{ij}[D_{max}(X_I, X_J) - D(X_i, X_j)] \\ & -(1-\delta)\sum_{i\in I}\sum_{j\in J} \gamma_{ij}[D(X_i, X_j) - D_{min}(X_I, X_J)]\,, \end{aligned} \tag{A.1}$$

donde

$$D_{max}(X_I, X_J) = \max_{i\in I}\max_{j\in J} D(X_i, X_j)$$

y

$$D_{min}(X_I, X_J) = \min_{i\in I}\min_{j\in J} D(X_i, X_j)\,.$$

La tabla A.1 muestra los valores de los parámetros α_{ij}, $\beta_{ii'}$, $\beta_{jj'}$, γ_{ij} y δ de la fórmula dada en la ecuación A.1, para los siete métodos de clustering jerárquico aglomerativo estudiados en el capítulo.

Resumiendo las ventajas principales de la nueva propuesta, éstas se pueden exponer en los siguientes puntos: i) cuando no aparecen empates, el algoritmo de grupo variable da el mismo resultado que el de grupo par; ii) el nuevo algoritmo siempre da una única solución; iii) en la representación de los resultados mediante multidendrogramas se puede apreciar explícitamente la aparición de empates durante el proceso de aglomeración y, además, la altura de todo intervalo de fusión indica el grado de heterogeneidad existente dentro del cluster correspondiente; iv) cuando existen empates, el algoritmo de grupo variable

es computacionalmente más eficiente que obtener todas las soluciones posibles deshaciendo los distintos empates en el algoritmo de grupo par; v) la nueva propuesta también se puede calcular de manera recursiva utilizando una generalización para la fórmula de Lance y Williams.

A pesar de que en los datos originales de proximidades no aparezcan empates, éstos pueden surgir durante el proceso de aglomeración. Por ese motivo, y dado que los resultados del algoritmo de grupo-variable coinciden con los del algoritmo de grupo par cuando no hay empates, recomendamos usar directamente la opción de grupo variable. De esta forma, en una única acción se averigua si existen empates o no, y además se obtiene la consiguiente solución.

Redes financieras y carteras

Un tipo particular de redes complejas son las redes basadas en correlaciones, es decir, redes utilizadas para visualizar la estructura de correlaciones entre un conjunto de variables. Concretamente, a partir de un conjunto de variables se puede calcular el coeficiente de correlación entre todos los pares. Al identificar las diferentes variables con los nodos de una red, todo par de nodos se puede conectar mediante un arco cuyo peso esté relacionado con el coeficiente de correlación entre las dos variables respectivas. Una red así construida es, por consiguiente, una red completamente conexa. Ejemplos bien conocidos de redes basadas en correlaciones se pueden encontrar en cualquier cartera de valores de un mercado financiero, al analizar la evolución de las series temporales obtenidas a partir de la diferencia diaria del precio de cierre de los valores.

En el capítulo 3 hemos descrito la complejidad de dichas redes financieras basadas en correlaciones. En estos casos, algunas veces es necesario filtrar dichas redes complejas para obtener subredes relevantes y más simples. En [60], Mantegna detectó una estructura jerárquica presente en una cartera de valores de un mercado financiero. El objetivo del estudio era conseguir la taxonomía de una cartera de valores utilizando para ello solamente la información de las series temporales de los precios de los valores. Partiendo de la matriz del coeficiente de correlación para un conjunto de valores bursátiles, se puede obtener una distancia métrica e identificar los grupos de compañías por medio del *árbol de recubrimiento mínimo* (ver figura A.2), que es equivalente al árbol jerárquico resultante del enlazado sencillo en términos de obtener la *ultramétrica subdominante* [77].

Con el propósito de analizar las diferencias en términos de distorsión del espacio que existen entre los *árboles financieros* obtenidos mediante distintos métodos de clustering jerárquico, hemos utilizado un conjunto de datos reales tomados del índice S&P 500. Solamente hemos tenido en cuenta los métodos aglomerativos que no producen inversiones en sus árboles jerárquicos, y hemos organizado los resultados en orden de distorsión del espacio. En un extremo de esta clasificación (ver figura A.3) aparece el método de enlazado sencillo, el cual sufre del conocido efecto de encadenamiento, mientras que en el otro extremo nos encontramos con los métodos de enlazado completo e inter-intra, donde ambos muestran estructuras internas claras en las ramas de sus árboles jerárquicos.

Otra importante aplicación de las técnicas de clustering jerárquico puede hallarse en la optimización de una cartera de valores. Desde la publicación del

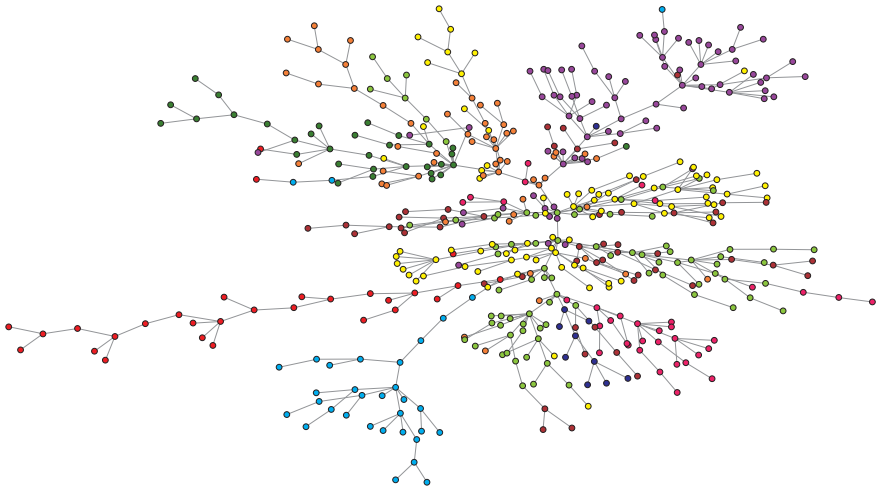


Figure A.2: Árbol de recubrimiento mínimo correspondiente a una cartera de valores del índice bursátil S&P 500. Los valores bursátiles están dibujados con distintos colores de acuerdo con los diez sectores industriales definidos por el GICS (*Global Industry Classification Standard*).

trabajo ya clásico de Markowitz [61], muchos otros trabajos se han dedicado a diversos aspectos de la optimización de carteras. En este capítulo también nos centramos en el problema de optimizar una cartera de valores. Trabajamos con una generalización del modelo estándar media-varianza de Markowitz en el que se incluyen restricciones de cardinalidad y de acotación. Estas restricciones por un lado garantizan la inversión en un número determinado de valores distintos, y por otra parte limitan la cantidad de capital a invertir en cada valor. Al considerar este modelo, el problema de optimizar una cartera se convierte en un problema mixto de programación cuadrática y entera, y, por consiguiente, no hay ningún método exacto capaz de resolver el problema de una manera eficiente. En una primera propuesta, mostramos cómo los árboles financieros, además de tener la habilidad de proporcionar clusters llenos de significado económico, también nos pueden ayudar en la optimización de una cartera. La propuesta consiste en dividir el problema en varios subproblemas según los distintos clusters que aparezcan en los árboles financieros. Hemos utilizado los dos métodos jerárquicos más prometedores para esta estrategia, esto es, el de enlazado completo y el inter-intra. A la vista de los resultados mostrados en la figura A.4, podemos concluir que el método de enlazado completo proporciona particiones de clusters que son generalmente más adecuadas para dividir y resolver el problema de optimizar una cartera mediante nuestro nuevo enfoque de clustering jerárquico.

En una segunda aproximación, hemos desarrollado un método heurístico basado en la red neuronal de Hopfield. Primero hemos recordado la forma particular de la función de energía de Hopfield, que es muy parecida a la función objetivo para optimizar una cartera de valores. A continuación hemos descrito la dinámica de la red y hemos analizado cómo satisfacer cada restricción del

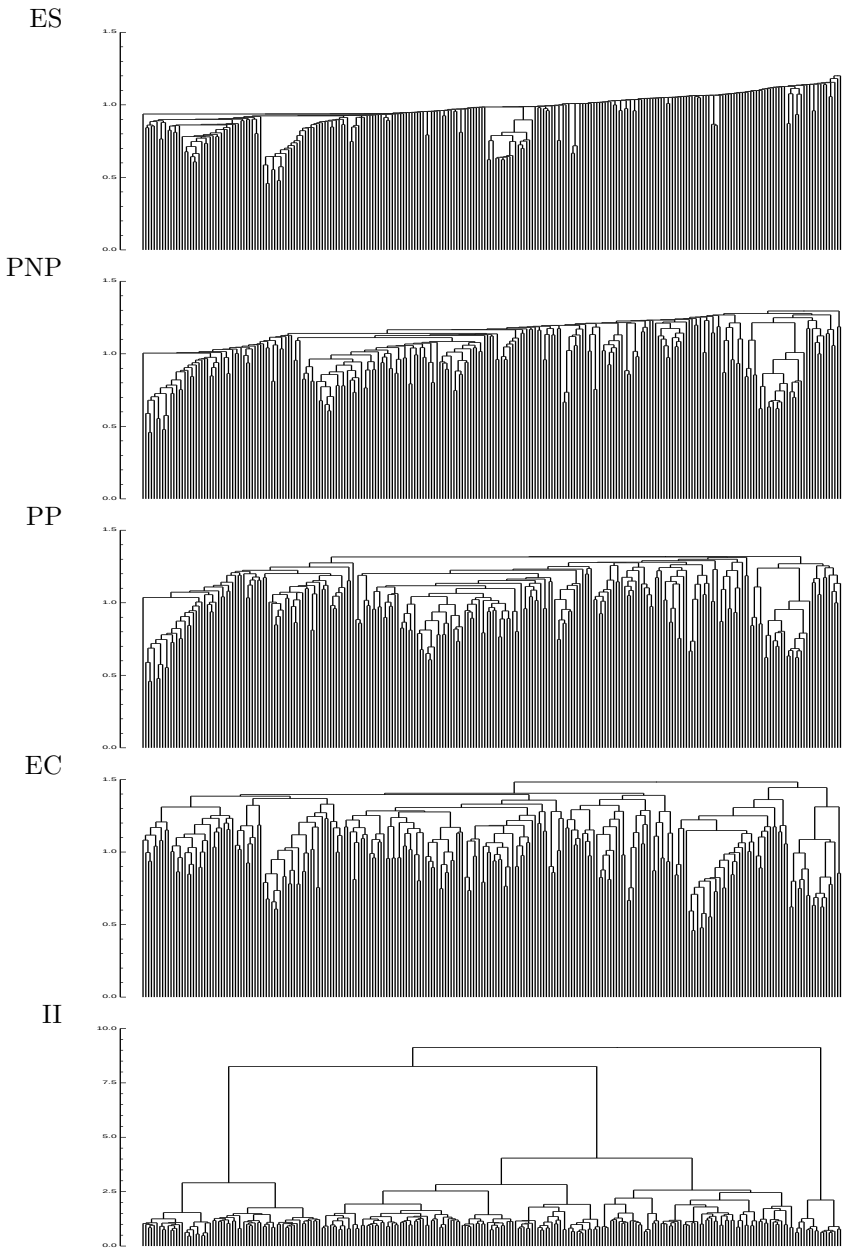


Figure A.3: Árboles financieros jerárquicos (multidendrogramas) para el el conjunto de prueba con $N = 250$ valores correspondientes al índice S&P 500. Los árboles están dispuestos, de arriba a abajo, en orden creciente de distorsión del espacio: Enlazado Sencillo (ES), Promedio No Pesado (PNP), Promedio Pesado (PP), Enlazado Completo (EC), e Inter-Intra (II).

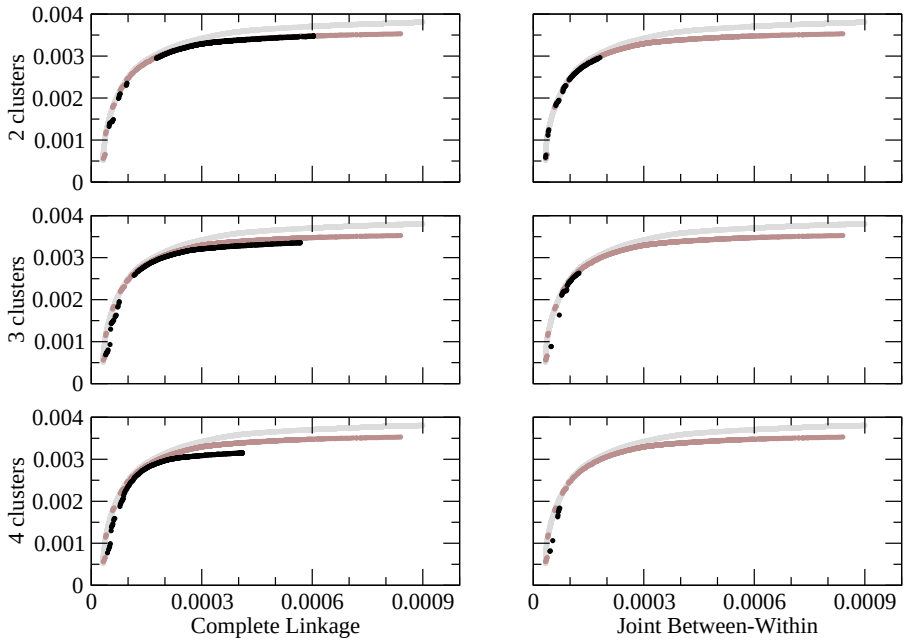


Figure A.4: Contribuciones a las fronteras de eficiencia fusionadas para los datos de prueba del índice S&P 500. En el eje x se representa la varianza de la rentabilidad y en el eje y se representa la rentabilidad media. Las fronteras de eficiencia estándar y general se muestran en colores más claros, mientras que las contribuciones a las fronteras de eficiencia fusionadas están pintadas de negro.

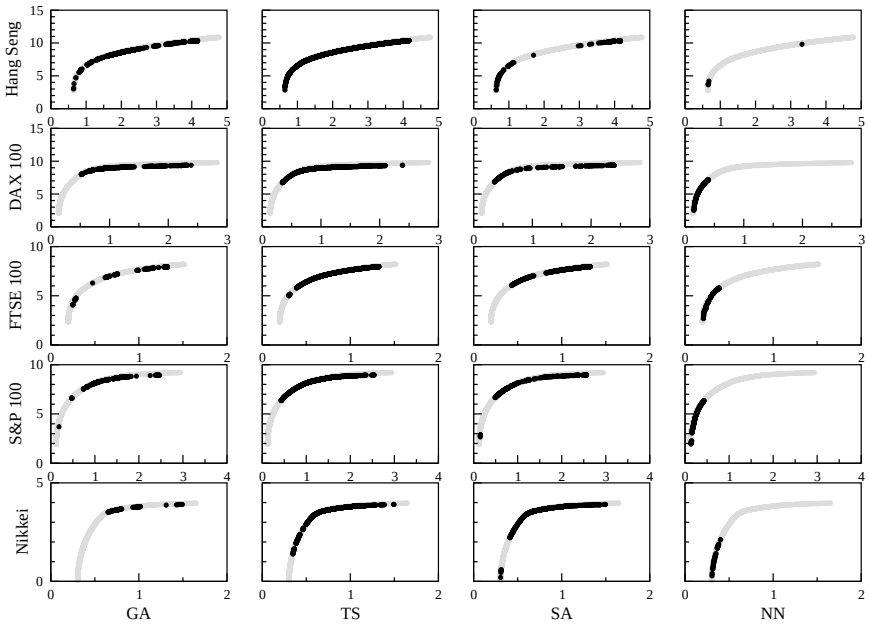


Figure A.5: Contribuciones a las fronteras de eficiencia fusionadas para cinco conjuntos de prueba. En el eje x se representa la varianza de la rentabilidad ($\times 10^3$) y en el eje y se representa la rentabilidad media ($\times 10^3$). Las fronteras de eficiencia estándar se muestran en gris, mientras que las contribuciones a las fronteras de eficiencia fusionadas están pintadas de negro.

problema. Finalmente hemos tomado los resultado obtenidos con nuestra red neuronal y los hemos comparado con los resultados obtenidos usando otros tres métodos heurísticos, a saber, algoritmos genéticos, búsqueda tabú y temple simulado. Todos los resultados experimentales presentados en este trabajo (ver a modo de resumen la figura A.5) nos conducen a concluir que ninguno de los cuatro métodos heurísticos utilizados ha sido mejor que el resto en todos los tipos de políticas de inversión. Sin embargo hemos observado que, al tratar con ejemplares del problema que requerían carteras verdaderamente diversificadas con un riesgo de inversión bajo, el modelo de red neuronal ha dado mejores soluciones que los otros tres métodos heurísticos.

Reducción del tamaño de redes complejas

Existen dos enfoques principales a la hora de realizar el paso de agrupamiento en cualquier procedimiento de clustering. Uno puede utilizar algoritmos jerárquicos y obtener una serie de particiones anidadas, o utilizar algoritmos particionales que originan una única partición de los datos. A pesar de sus diferencias, ambas técnicas comparten un punto en común en la posible utilización de una *función de calidad*, esto es, un criterio cuantitativo para evaluar cuán buenas son las particiones. En el clustering jerárquico dicha función se necesita para saber cuál de las particiones de la jerarquía es la mejor, y el clustering particional

habitualmente produce clusters optimizando una función de calidad.

En términos de cómputo, el problema de detección de comunidades es muy semejante al problema de encontrar un estado fundamental de un modelo de vidrio de espín. Un vidrio de espín es un material desordenado que exhibe una elevada frustración magnética debido a la incapacidad del sistema para permanecer en un único estado de energía mínima (el estado fundamental). Reichardt y Bornholdt [79] establecieron la base para un marco unificado bajo el cual se puede incluir la detección de comunidades. Ellos mostraron que dicho problema se puede hacer corresponder con el problema de hallar el estado fundamental de un vidrio de espín de Potts de rango infinito, donde los valores de similitud se traducen en fuerzas de conexiones, la energía del sistema se interpreta como la función de calidad de la partición en comunidades, y los estados de los espines son los índices de las comunidades.

Hemos comenzado el capítulo 4 describiendo una de las funciones de calidad más populares para la detección de comunidades, la *modularidad*, bajo un marco común para funciones de calidad provenientes de un modelo particular de vidrio de espín. La modularidad fue propuesta por Newman y Girvan [69] como una función para calcular la calidad de las particiones existentes a lo largo de un dendrograma, y poder buscar óptimos locales que indicarían particiones satisfactorias. Dada una red particionada en comunidades, siendo C_i la comunidad del nodo i, la definición matemática de modularidad se puede expresar en términos de la matriz de adyacencia pesada, w_{ij}, que representa el valor del peso en el arco entre i y j, como:

$$Q = \frac{1}{2w} \sum_i \sum_j \left(w_{ij} - \frac{w_i w_j}{2w} \right) \delta(C_i, C_j) \,, \tag{A.2}$$

donde el peso del nodo i es $w_i = \sum_j w_{ij}$, el peso total de la red es $2w = \sum_i w_i = \sum_i \sum_j w_{ij}$, y la función delta de Kronecker, $\delta(C_i, C_j)$, toma el valor 1 si los nodos i y j están en la misma comunidad y el valor 0 en otro caso.

El reto de optimizar la modularidad ha merecido muchos esfuerzos por parte de la comunidad científica durante estos últimos años. Dado que el problema tiene un coste computacional NP, no se puede llevar a cabo mediante búsqueda exhaustiva y solamente las heurísticas de optimización han demostrado ser competentes a la hora de buscar soluciones subóptimas de la función de modularidad en un tiempo de cálculo razonable. Aquí hemos propuesto un método exacto que permite transformar una red determinada en otra más pequeña que conserva el mismo valor de modularidad, independientemente de la partición considerada. Primero hemos descrito la propiedad de la modularidad que permite la reducción de tamaño de redes complejas, a saber, que los nodos que forman una comunidad en cualquier partición óptima se pueden representar mediante un único nodo en una red reducida. Toda la información necesaria para calcular la modularidad, cada nodo de la red reducida la reúne en su auto-loop (que representa la intraconectividad de la comunidad) y en sus arcos (que representan la interconectividad con el resto de la red).

A continuación hemos presentado dos posibles reducciones analíticas que preservan la modularidad en redes pesadas (tanto dirigidas como no dirigidas): *pelos*, esto es, nodos conectados a la red mediante un único arco; y *pelos triangulares*, que son estructuras particulares compuestas por tres nodos (ver figura A.6). Finalmente hemos estimado la cantidad de reducciones de tamaño (es de-

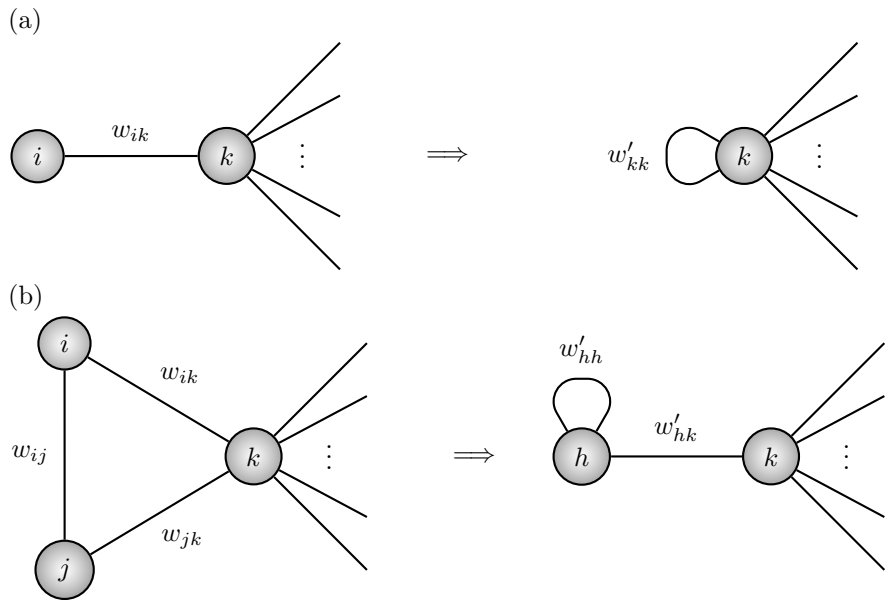


Figure A.6: Reducciones analíticas para redes no dirigidas: (a) ejemplo de reducción de un *pelo*; (b) ejemplo de reducción de un *pelo triangular*. El caso particular de redes no pesadas, en las que todos los pesos valen 1, implica $w'_{kk} = 2$ en la reducción (a), e implica $w'_{hh} = 2$ y $w'_{hk} = 2$ en la reducción (b).

cir, el número de pelos) que uno puede esperar conseguir para las distribuciones de grado más habituales en redes complejas: sin escala y exponencial.

El uso sistemático de esta reducción de tamaño permite realizar búsquedas más exhaustivas a través del espacio de particiones, la cual cosa se traducirá normalmente en mejores valores de modularidad comparamos con los obtenidos sin usar la reducción de tamaño. Creemos que la idea de la reducción de tamaño analítica se podría extender a otros motifs (bloques constituyentes) de las redes, aunque su tratamiento analítico es posible que sea más difícil.

Niveles de resolución en redes complejas

Volviendo al ejemplo de la cartera con valores bursátiles del índice S&P 500, en la figura A.2 mostrábamos los valores bursátiles clasificados en 10 grupos según los distintos sectores definidos por el GICS (*Global Industry Classification Standard*): energía, materiales, industrial, productos de consumo no básico, productos de primera necesidad, salud, finanzas, tecnología de la información, servicios de telecomunicaciones, y servicios de utilidad pública. Sin embargo, esta clasificación es solamente una de las muchas posibles. De hecho, el mismo GICS ofrece tres niveles más de clasificación para la industria. Así, uno puede clasificar una cartera de valores en: 10 sectores, 24 grupos de industrias, 59 industrias, o 112 ramas industriales.

La existencia de varias escalas de descripción no es simplemente una peculiaridad de los sistemas financieros, sino que es una característica común a

muchos sistemas complejos reales. En el capítulo 5, motivados por el reciente hallazgo de que la optimización de la modularidad presenta un límite de resolución relacionado con la escala característica impuesta por la suma de pesos total de la red, hemos presentado un método de resolución múltiple con el que se consigue que la optimización de la modularidad pueda hallar estructura de comunidades en distintas escalas de descripción. La idea principal consiste en reescalar la topología definiendo una nueva red a partir de la original, añadiendo a cada nodo un auto-loop de igual magnitud que hemos llamado *resistencia*. La nueva red presenta las mismas características que la red original en términos de conectividad, pero permite buscar módulos a través de diferentes escalas topológicas.

Con el fin de hallar la estructura de comunidades, hemos propuesto un nuevo algoritmo para optimizar la modularidad basado en búsqueda tabú. La principal virtud de este algoritmo es que se trata de una combinación de procesos aglomerativos y divisivos, evitando así los inconvenientes que presentan ambas estrategias por separado. Además, el proceso iterativo puede comenzar desde cualquier partición inicial, lo cual es provechoso para el cálculo de la mesoescala, puesto que las particiones óptimas para valores próximos de la resistencia acostumbran a ser muy similares. Como validación del método, hemos proporcionado ejemplos de la subestructura y superestructura modular hallada en diversas redes complejas, tanto reales como sintéticas (ver figura A.7). Los resultados son conjuntos de particiones que muestrean, en cada escala particular, el rango de módulos estructurales desde los nodos individualmente hasta toda la red al completo.

También hemos presentado un método para desvelar información a partir de las diversas escalas de descripción que aparecen en muchos sistemas complejos reales. Obtenemos una representación gráfica de la mesoescala al completo superponiendo todas las escalas halladas, ponderadas por el intervalo en la resistencia durante el cual prevalecen. El resultado es una *matriz de mesoescalas* cuya representación proporciona un mapa estructural de la topología de la red. Las matrices de mesoescalas para las redes sintéticas mostradas en la figura A.8 revelan cómo los nodos forman grupos a diferentes escalas. Sin embargo, las simetrías presentes en las redes analizadas juegan en favor de esta clara visualización. En redes complejas reales, estas simetrías están habitualmente ausentes y es necesario un proceso de filtrado para poder revelar esta misma información. Con este propósito hemos diseñado lo que llamamos la *matriz de mesoescalas filtrada*, que consiste en: i) fijar una mesoescala (nivel de color) y quitar de la matriz de mesoescalas los elementos por debajo de dicho nivel (colores más claros); ii) calcular las componentes conexas de los elementos restantes (grupos); y iii) reordenar la matriz de izquierda a derecha en orden de tamaño decreciente, respetando los grupos obtenidos en niveles anteriores. Este proceso se repite desde la mesoescala más baja hasta la más alta, acumulando los resultados de las etapas previas. De este modo, se consigue una representación más clara del mapa estructural sin perder ninguna información de la matriz de mesoescalas original.

Hemos aplicado este método con el fin de desvelar las mesoescalas correspondientes a la conectividad neuronal del *nematodo C. elegans*. El sistema nervioso completo de este nematodo se puede representar mediante una red de adyacencia. La matriz de mesoescalas obtenida en este caso es difícil de analizar porque: primero, el orden de los nodos no está prescrito; y, segundo, los grupos

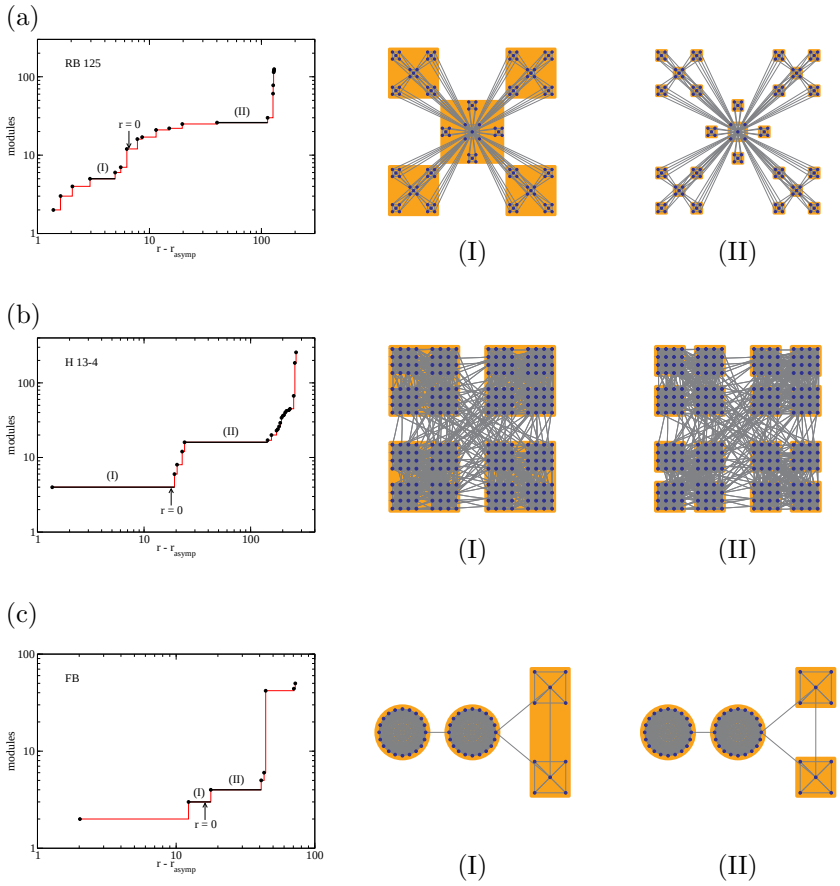


Figure A.7: Resolución múltiple de la estructura modular en redes sintéticas. Izquierda: número de módulos en la partición óptima obtenida para la modularidad, donde cada punto corresponde a una partición diferente y las flechas señalan las particiones óptimas en el valor 0 de resistencia. Derecha: redes analizadas, destacando las particiones correspondientes a dos escalas representativas indicadas por (I) y (II). (a) RB 125 es una extensión de la red jerárquica propuesta en [78]. Las regiones correspondientes a 5, 25 y 26 módulos son las más representativas (estables) en términos de resolución. (b) H 13-4 es una red homogénea en grado y con dos niveles jerárquicos predefinidos. Ambos niveles son descubiertos en distintas escalas. (c) FB es la red propuesta en [32] para demostrar el límite de resolución de la modularidad. Este límite es superado en la escala (II), obteniendo la partición esperada.

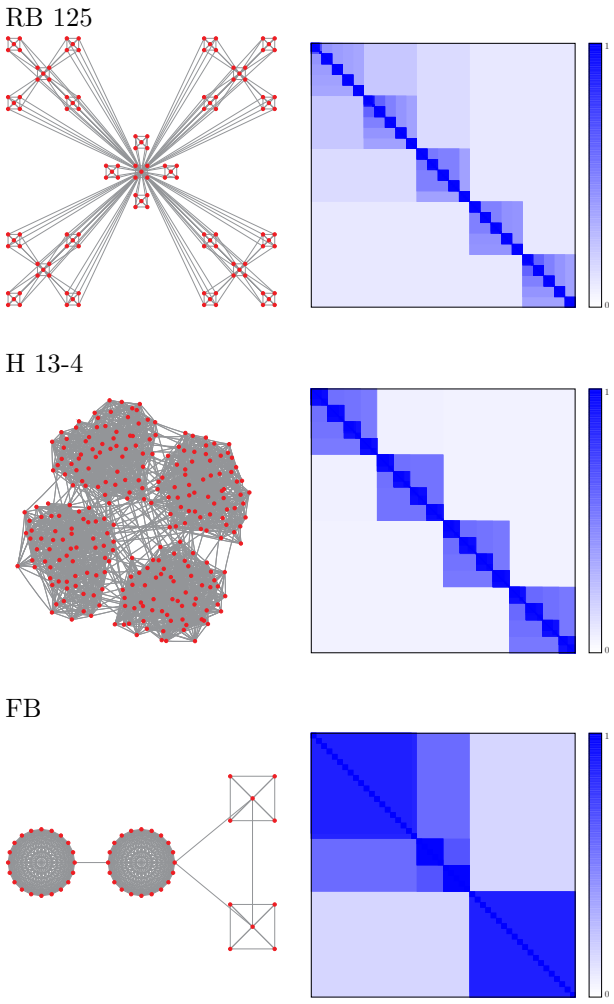


Figure A.8: Redes complejas sintéticas y sus matrices de mesoescalas. Los niveles de color se corresponden con la persistencia de las estructuras para distintos valores de la resistencia. Dibujamos tanto las redes (izquierda) como sus respectivas matrices de mesoescalas (derecha). RB 125 corresponde a una extensión de la red jerárquica propuesta en [78]. H 13-4 corresponde a una red homogénea en grado y con dos niveles jerárquicos predefinidos. FB corresponde a la red propuesta en [32] para demostrar el límite de resolución de la modularidad.

obtenidos en escalas distintas no son todos jerárquicos. Para realzar la visualización hemos calculado la correspondiente matriz de mesoescalas filtrada. El análisis de la matriz de mesoescalas filtrada del *C. elegans* muestra varias correlaciones interesantes de las subestructuras que prevalecen en las mesoescalas, por una parte con la localización del soma de las neuronas, y por otra con las funcionalidades en el gusano. Estos resultados pueden ser útiles para los biólogos a la hora de diseñar experimentos con objetivos específicos basados en la clasificación de las neuronas según sus funciones en las diferentes escalas topológicas.

En la última sección de este capítulo hemos presentado una discusión acerca del papel de las distintas escalas topológicas más allá de su definición estática, revelando su implicación en procesos dinámicos que hay sobre las redes. También hemos comparado nuestro método con otros posibles acercamientos a la mesoescala. Y finalmente hemos dado una perspectiva sobre el significado de las mesoescalas comparándolo con la comúnmente aceptada escala única de descripción.

El estudio de distintas escalas de descripción debería ser útil para comprender mejor las redes complejas. El análisis de los resultados revela que algunas escalas topológicas son más persistentes (estables) que otras, en términos de resolución. Estas escalas más estables proporcionan información específica sobre los principales aspectos modulares de la estructura: en las redes sintéticas analizadas, se corresponden con las escalas estructurales predefinidas impuestas *ad hoc*; y en las redes reales, se corresponden exactamente con conocimiento previo sobre las redes que hasta el momento no ha sido recuperado por ningún otro método que haya estudiado estas redes. Con nuestro método, la optimización de la modularidad queda liberada de problemas de resolución, y damos nuevas ideas acerca de la descripción de redes complejas. La existencia de varias escalas de descripción en redes complejas presenta muchas analogías con el estudio de sistemas complejos en física, donde se han formulado diversos modelos en distintas escalas espaciales para poder comprender diferentes fenómenos.

Descripciones generales de comunidades

El análisis de la estructura modular utilizando la modularidad como función de calidad proporciona una partición de la red en comunidades, donde cada comunidad es un subconjunto de nodos más conectados entre ellos que con el resto de nodos de la red. Sin embargo, la modularidad se basa enormemente de comunidades y, por consiguiente, no se puede utilizar para detectar grupos generales de nodos revelados mediante patrones de conectividad alternativos. A pesar de que se ha trabajado mucho con el fin de obtener técnicas fiables para optimizar la modularidad, hasta ahora se ha hecho muy poco por analizar el concepto de modularidad propiamente dicho y su fiabilidad como método para la detección de estructura modular más general.

En el capítulo 6 hemos propuesto un marco general para describir grupos de nodos en redes utilizando *motifs* (pequeñas subredes conexas) como unidades elementales. En concreto, hemos dado varias definiciones de grupos de nodos, incluyendo las comunidades, basadas en el principio de que contengan más motifs que un modelo de caso nulo que representa una versión aleatorizada de la red en estudio. De este modo, hemos desarrollado la formulación matemática

para extensiones a la modularidad donde los bloques constituyentes son distintos tipos de motifs (por ejemplo triángulos, ciclos y caminos entre nodos), y no simplemente arcos como es el caso de la expresión original de la modularidad. Primero hemos explicado el marco más general de la modularidad basada en motifs, que por analogía con la modularidad estándar hemos definido como la fracción de motifs que hay dentro de las comunidades menos la fracción que habría en una red aleatoria que preservase los grados de los nodos:

$$Q_{\mathcal{M}}(C) = \frac{\Psi_{\mathcal{M}}(C)}{\Psi_{\mathcal{M}}} - \frac{\Omega_{\mathcal{M}}(C)}{\Omega_{\mathcal{M}}}\,, \tag{A.3}$$

donde $\Psi_{\mathcal{M}}(C)$ es el número de motifs incluidos de forma completa dentro de las comunidades que forman una partición determinada C, y su valor máximo, $\Psi_{\mathcal{M}}$, se corresponde con la partición en una única comunidad que contenga todos los nodos. Para una red aleatoria que preserve los grados de los nodos, estas mismas cantidades son respectivamente $\Omega_{\mathcal{M}}(C)$ y $\Omega_{\mathcal{M}}$. A continuación hemos aplicado este formalismo a varios tipos de motifs. Entre los motifs más simples, los triángulos han recibido mucha atención en los trabajos sobre redes complejas (por ejemplo, en la definición del coeficiente de clustering). Por este motivo, nuestro primer paso ha consistido en definir la modularidad basada en triángulos, que nos permite hallar comunidades hechas a base de triángulos. Además, el hecho de que los triángulos se pueden generalizar fácilmente a ciclos, nos ha conducido inmediatamente a la fórmula de la modularidad basada en ciclos. Por último, hemos mostrado que la modularidad basada en caminos también es posible. De hecho, los caminos de longitud 2 han demostrado ser muy útiles en el estudio de redes bipartitas.

Hemos probado estas nuevas versiones de modularidad con redes sintéticas y reales (ver figura A.9). Utilizando el tipo de motif que era pertinente en cada caso, hemos podido recuperar los patrones de conectividad esperados en las redes, tanto para el caso de redes con estructura modular como para el caso de redes con estructura multipartita. Además, este principio va más allá del uso de la modularidad y creemos que podría inspirar otros marcos alternativos igualmente prometedores con el fin de detectar grupos genéricos de nodos en redes.

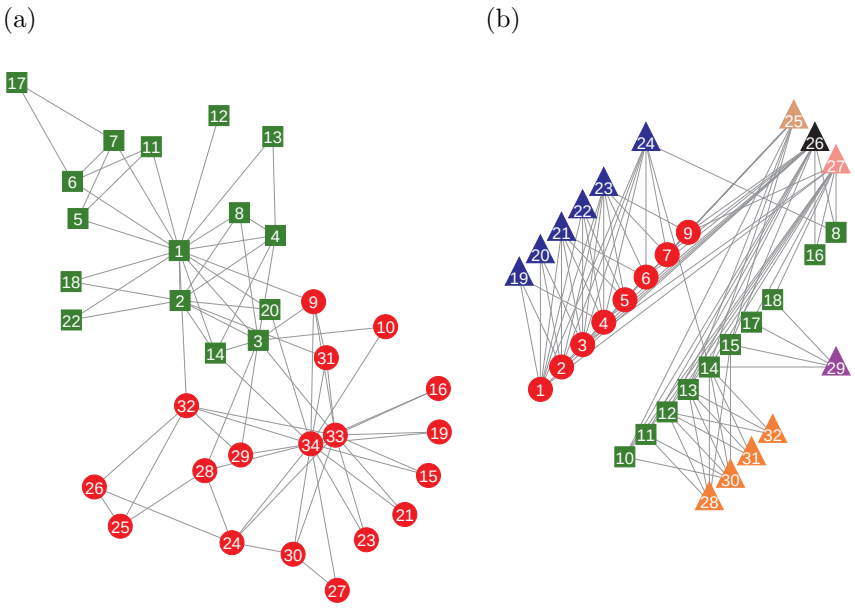


Figure A.9: Resultados obtenidos utilizando modularidad basada en motifs para dos redes reales: (a) red del club de karate Zachary, donde representamos la división real obtenida utilizando distintas modularidades basadas en caminos y ciclos; (b) red bipartita que representa la participación en eventos por mujeres, donde los resultados (que muestran claramente la diferencia de roles entre mujeres y eventos) han sido obtenidos aplicando modularidad basada en caminos de longitud 2 con nodos intermedios libres.

Appendix B

List of Publications

- A. Fernández and S. Gómez.
  Solving non-uniqueness in agglomerative hierarchical clustering using multidendrograms.
  Journal of Classification, 25:43–65, 2008.

- A. Fernández and S. Gómez.
  Portfolio selection using neural networks.
  Computers & Operations Research, 34:1177–1191, 2007.

- A. Arenas, J. Duch, A. Fernández, and S. Gómez.
  Size reduction of complex networks preserving modularity.
  New Journal of Physics, 9:176, 2007.

- A. Arenas, A. Fernández, and S. Gómez.
  Analysis of the structure of complex networks at different resolution levels.
  New Journal of Physics, 10:053039, 2008.

- A. Arenas, A. Fernández, and S. Gómez.
  A complex network approach to the determination of functional groups in the neural system of *C. elegans*.
  Lecture Notes in Computer Science, in press, 2008.

- A. Arenas, A. Fernández, S. Fortunato, and S. Gómez.
  Motif-based communities in complex networks.
  Journal of Physics A: Mathematical and Theoretical, 41:224001, 2008.

Bibliography

[1] L.A. Adamic and E. Adar. Friends and neighbors on the web. *Social Networks*, 25(3):211–230, 2003.

[2] R. Albert, H. Jeong, and A.L. Barabási. Diameter of the world-wide web. *Nature*, 401:130–131, 1999.

[3] A. Arenas, A. Díaz-Guilera, and C.J. Pérez-Vicente. Synchronization processes in complex networks. *Physica D: Nonlinear Phenomena*, 224(1–2):27–34, 2006.

[4] A. Arenas, A. Díaz-Guilera, and C.J. Pérez-Vicente. Synchronization reveals topological scales in complex networks. *Physical Review Letters*, 96:114102, 2006.

[5] A. Arenas, J. Duch, A. Fernández, and S. Gómez. Size reduction of complex networks preserving modularity. *New Journal of Physics*, 9:176, 2007.

[6] A. Arenas, A. Fernández, S. Fortunato, and S. Gómez. Motif-based communities in complex networks. *Journal of Physics A: Mathematical and Theoretical*, 41:224001, 2008.

[7] A. Arenas, A. Fernández, and S. Gómez. Analysis of the structure of complex networks at different resolution levels. *New Journal of Physics*, 10:053039, 2008.

[8] A. Arenas, A. Fernández, and S. Gómez. A complex network approach to the determination of functional groups in the neural system of *C. Elegans*. *Lecture Notes in Computer Science*, in press, 2008.

[9] V. Arnau, S. Mars, and I. Marín. Iterative cluster analysis of protein interaction data. *Bioinformatics*, 21(3):364–378, 2005.

[10] T. Backeljau, L. De Bruyn, H. De Wolf, K. Jordaens, S. Van Dongen, and B. Winnepenninckx. Multiple UPGMA and neighbor-joining trees and the performance of some computer packages. *Molecular Biology and Evolution*, 13(2):309–313, 1996.

[11] A.-L. Barabási. Network theory–the emergence of the creative enterprise. *Science*, 308:639–641, 2005.

[12] M.J. Barber. Modularity and community detection in bipartite networks. Preprint arXiv:0707.1616, 2007.

[13] E.T. Bell. Exponential numbers. *Amer. Math. Monthly*, 41:411–419, 1934.

[14] M. Bernaschi, L. Grilli, and D. Vergni. Statistical analysis of fixed income market. *Physica A*, 308:381–390, 2002.

[15] M. Boguna, R. Pastor-Satorras, A. Díaz-Guilera, and A. Arenas. Models of social networks based on social distance attachment. *Physical Review E*, 70(056122), 2004.

[16] G. Bonanno, N. Vandewalle, and R.N. Mantegna. Taxonomy of stock market indices. *Physical Review E*, 62(6):R7615–R7618, 2000.

[17] U. Brandes, D. Delling, M. Gaertler, R. Görke, M. Hoefer, Z. Nikoloski, and D. Wagner. On finding graph clusterings with maximum modularity. In *Proceedings of the 33rd International Workshop on Graph-Theoretical Concepts in Computer Science (WG'07)*, Berlin-Heidelberg, Germany, 2007. Springer Verlag.

[18] T.-J. Chang, N. Meade, J.E. Beasley, and Y.M. Sharaiha. Heuristics for cardinality constrained portfolio optimisation. *Computers & Operations Research*, 27(13):1271–1302, 2000.

[19] A. Clauset, M.E.J. Newman, and C. Moore. Finding community structure in very large networks. *Physical Review E*, 70(066111), 2004.

[20] R.M. Cormack. A review of classification (with discussion). *Journal of the Royal Statistical Society, Ser. A*, 134:321–367, 1971.

[21] L. Danon, A. Díaz-Guilera, and A. Arenas. The effect of size heterogeneity on community identification in complex networks. *Journal of Statistical Mechanics: Theory and Experiment*, November(P11010), 2006.

[22] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas. Comparing community structure identification. *Journal of Statistical Mechanics: Theory and Experiment*, September(P09008), 2005.

[23] A. Davis, B.B. Gardner, and M.R. Gardner. *Deep South*. Chicago: The University of Chicago Press, 1941.

[24] J.L. Dubien and W.D. Warde. A mathematical comparison of the members of an infinite family of agglomerative clustering algorithms. *Canadian Journal of Statistics*, 7:29–38, 1979.

[25] J. Duch and A. Arenas. Community identification using extremal optimization. *Physical Review E*, 72(027104), 2005.

[26] R.M. Durbin. *Studies on the Development and Organisation of the Nervous System of Caenorhabditis Elegans*. PhD thesis, University of Cambridge, 1987.

[27] K. Eriksen, I. Simonsen, S. Maslov, and K. Sneppen. Modularity and extreme edges of the internet. *Physical Review Letters*, 90(148701), 2003.

[28] A. Fernández and S. Gómez. Portfolio selection using neural networks. *Computers & Operations Research*, 34:1177–1191, 2007.

[29] A. Fernández and S. Gómez. Solving non-uniqueness in agglomerative hierarchical clustering using multidendrograms. *Journal of Classification*, 25:43–65, 2008.

[30] J. Fieldsend, J. Matatko, and M. Peng. Cardinality constrained portfolio optimisation. In *Proceedings of the Fifth International Conference on Intelligent Data Engineering and Automated Learning (IDEAL'2004)*, Exeter, Aug 2004.

[31] G.W. Flake, S. Lawrence, C.L. Giles, and F.M. Coetzee. Self-organization and identification of web communities. *IEEE Computer*, 35(3):66–71, 2002.

[32] S. Fortunato and M. Barthélemy. Resolution limit in community detection. *Proceedings of the National Academy of Science of the USA*, 104(1):36–41, 2007.

[33] L. Freeman. Finding social groups: A meta-analysis of the southern women data. In K. Carley R. Breiger and P. Pattison, editors, *Dynamic Social Network Modeling Analysis: Workshop Summary and Papers*, page 39, Washington DC, 2003. The National Academies Press.

[34] M. Gilli and E. Këllezi. Heuristic approaches for portfolio optimization. In *Sixth International Conference on Computing in Economics and Finance of the Society for Computational Economics*, Barcelona, Jul 2000.

[35] M. Girvan and M.E.J. Newman. Community structure in social and biological networks. *Proceedings of the National Academy of Science of the USA*, 99(12):7821–7826, 2002.

[36] P. Gleiser and L. Danon. Community structure in jazz. *Adv. Complex Systems*, 6:565–, 2003.

[37] F. Glover. Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, 5:533–549, 1986.

[38] A.D. Gordon. *Classification*. Chapman & Hall/CRC, second edition, 1999.

[39] L. Grilli. Long-term fixed income market structure. *Physica A*, 332:441–447, 2004.

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

[41] R. Guimerà, L. Danon, A. Diaz-Guilera, F. Giralt, and A. Arenas. Self-similar community structure in a network of human interactions. *Physical Review E*, 68(065103), 2003.

[42] R. Guimerà, S. Mossa, A. Turtschi, and L.A.N. Amaral. The worldwide air transportation network: Anomalous centrality, community structure, and cities' global roles. *Proceedings of the National Academy of Sciences of the United States of America*, 102(22):7794–7799, 2005.

[43] R. Guimerà, M. Sales-Pardo, and L.A.N. Amaral. Classes of complex networks defined by role-to-role connectivity profiles. *Nature Phys.*, 3:63–, 2007.

[44] R. Guimerà, M. Sales-Pardo, and L.A.N. Amaral. Module identification in bipartite and directed networks. *Physical Review E*, 76(036102), 2007.

[45] G. Hart. The occurrence of multiple UPGMA phenograms. In J. Felsenstein, editor, *Numerical Taxonomy*, pages 254–258. Springer-Verlag Berlin, Heidelberg, 1983.

[46] P. Holme, M. Huss, and H. Jeong. Subnetwork hierarchies of biochemical pathways. *Bioinformatics*, 19:532–, 2003.

[47] J.J. Hopfield. Neurons with graded response have collective computational properties like those of two-state neurons. *Proceedings of the National Academy of Sciences*, 81(10):3088–3092, 1984.

[48] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice-Hall, New York, 1988.

[49] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: A review. *ACM Computing Surveys*, 31(3):264–323, 1999.

[50] N.J. Jobst, M.D. Horniman, C.A. Lucas, and G. Mitra. Computational aspects of alternative portfolio selection models in the presence of discrete asset choice constraints. *Quantitative Finance*, 1(5):489–501, 2001.

[51] H. Kellerer and D. Maringer. Optimization of cardinality constrained portfolios with an hybrid local search algorithm. In *Fourth Metaheuristics International Conference (MIC'2001)*, Porto, Jul 2001.

[52] L. Kullmann, J. Kertész, and K. Kaski. Time-dependent cross-correlations between different stock returns: A directed network of influence. *Physical Review E*, 66(026125), 2002.

[53] J.M. Kumpula, J. Saramäki, K. Kaski, and J. Kertész. Resolution limit in complex network community detection with potts model approach. *The European Physical Journal B*, 56:41, 2007.

[54] L. Laloux, P. Cizeau, J.-P. Bouchaud, and M. Potters. Noise dressing of financial correlation matrices. *Physical Review Letters*, 83(7):1467–1470, 1999.

[55] G.N. Lance and W.T. Williams. A generalized sorting strategy for computer classifications. *Nature*, 212:218, 1966.

[56] D. Lin, S. Wang, and H. Yan. A multiobjective genetic algorithm for portfolio selection problem. In *Proceedings of ICOTA 2001*, Hong Kong, Dec 2001.

[57] D. Lusseau and M.E.J. Newman. Identifying the role that animals play in their social networks. *Proc. R. Soc. Lond. B*, 271:S477, 2004.

[58] D. Lusseau, K. Schneider, O.J. Boisseau, P. Haase, E. Slooten, and S.M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. can geographic isolation explain this unique trait? *Behav. Ecol. Sociobiol.*, 54:396, 2003.

[59] J. MacCuish, C. Nicolaou, and N.E. MacCuish. Ties in proximity and clustering compounds. *Journal of Chemical Information and Computer Sciences*, 41:134–146, 2001.

[60] R.N. Mantegna. Hierarchical structure in financial markets. *The European Physical Journal B*, 11(1):193–197, 1999.

[61] H. Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952.

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

[63] B.J.T. Morgan and A.P.G. Ray. Non-uniqueness and inversions in cluster analysis. *Applied Statistics*, 44(1):117–134, 1995.

[64] M.E.J. Newman. Analysis of weighted networks. *Physical Review E*, 70(056131), 2004.

[65] M.E.J. Newman. Detecting community structure in networks. *European Physical Journal B*, 38:321–330, 2004.

[66] M.E.J. Newman. Fast algorithm for detecting community structure in networks. *Physical Review E*, 69(066133), 2004.

[67] M.E.J. Newman. Finding community structure in networks using the eigenvectors of matrices. *Physical Review E*, 74(036104), 2006.

[68] M.E.J. Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences of the United States of America*, 103:8577–8582, 2006.

[69] M.E.J. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69(026113), 2004.

[70] M.E.J. Newman, D.J. Watts, and S.H. Strogatz. Random graph models of social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 99:2566–2572, 2002.

[71] J.-P. Onnela, A. Chakraborti, K. Kaski, and J. Kertész. Dynamic asset trees and portfolio analysis. *The European Physical Journal B*, 30:285–288, 2002.

[72] J.-P. Onnela, A. Chakraborti, K. Kaski, J. Kertész, and A. Kanto. Dynamics of market correlations: Taxonomy and portfolio analysis. *Physical Review E*, 68(5):056110, 2003.

[73] 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.

[74] C. Peterson and B. Söderberg. A new method for mapping optimization problems onto neural networks. *International Journal of Neural Systems*, 1(1):3–22, 1989.

[75] V. Plerou, P. Gopikrishnan, B. Rosenow, L.A.N. Amaral, and H.E. Stanley. Universal and nonuniversal properties of cross correlations in financial time series. *Physical Review Letters*, 83(7):1471–1474, 1999.

[76] J.M. Pujol, J. Béjar, and J. Delgado. Clustering algorithm for determining community structure in large networks. *Physical Review E*, 74(016107), 2006.

[77] R. Rammal, G. Toulouse, and M.A. Virasoro. Ultrametricity for physicists. *Reviews of Modern Physics*, 58(3):765–788, 1986.

[78] E. Ravasz and A.-L. Barabasi. Hierarchical organization in complex networks. *Physical Review E*, 67(026112), 2003.

[79] J. Reichardt and S. Bornholdt. Statistical mechanics of community detection. *Physical Review E*, 74(016110), 2006.

[80] M. Sales-Pardo, R. Guimerà, A. Moreira, and L.A.N. Amaral. Extracting the hierarchical organization of complex systems. *Proceedings of the National Academy of Sciences of the United States of America*, 104:15224–, 2007.

[81] Y. Sawaragi, H. Nakayama, and T. Tanino. Theory of multiobjective optimization. In R. Bellman, editor, *Mathematics in Science and Engineering*, volume 176. Academic Press Inc., New York, 1985.

[82] A. Schaerf. Local search techniques for constrained portfolio selection problems. *Computational Economics*, 20(3):177–190, 2002.

[83] K.A. Smith. Neural networks for combinatorial optimization: A review of more than a decade of research. *INFORMS Journal on Computing*, 11(1):15–34, 1999.

[84] P.H.A. Sneath and R.R. Sokal. *Numerical Taxonomy: The Principles and Practice of Numerical Classification*. W. H. Freeman and Company, 1973.

[85] C.M. Song, S. Havlin, and H.A. Makse. Self-similarity of complex networks. *Nature*, 433:392–395, 2005.

[86] F. Streichert, H. Ulmer, and A. Zell. Evolutionary algorithms and the cardinality constrained portfolio optimization problem. In *Selected Papers of the International Conference on Operations Research (OR'2003)*, Heidelberg, Sep 2003.

[87] S.H. Strogatz. Exploring complex networks. *Nature*, 410:268–276, 2001.

[88] G.J. Székely and M.L. Rizzo. Hierarchical clustering via joint between-within distances: Extending Ward's minimum variance method. *Journal of Classification*, 22:151–183, 2005.

[89] V. Tola, F. Lillo, M. Gallegati, and R.N. Mantegna. Cluster analysis for portfolio optimization. Preprint arXiv:physics/0507006 [physics.soc-ph], 2005.

[90] W.A. Van der Kloot, A.M.J. Spaans, and W.J. Heiser. Instability of hierarchical cluster analysis due to input order of the data: The PermuCLUSTER solution. *Psychological Methods*, 10(4):468–476, 2005.

[91] X. Wang, A. Jagota, F. Botelho, and M. Garzon. Absence of cycles in symmetric neural networks. *Neural Computation*, 10(5):1235–1249, 1998.

[92] Jr. Ward, J.H. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58:236–244, 1963.

[93] J.G. White, E. Southgate, J.N. Thompson, and S. Brenner. The structure of the nervous system of the nematode caenorhabditis elegans. *Phil. Trans. Royal Soc. London. Series B*, 314:1–340, 1986.

[94] Y. Xia, B. Liu, S. Wang, and K.K. Lai. A model for portfolio selection with order of expected returns. *Computers & Operations Research*, 27(5):409–422, 2000.

[95] W.W. Zachary. An information flow model for conflict and fission in small groups. *J. Anthr. Res.*, 33:452–, 1977.