SAPIENZA
UNIVERSITÀ DI ROMA

Analysis and modeling of complex networks by means of computational intelligence techniques

Dept. of Information Engineering, Electronics and Telecommunications

Dottorato di Ricerca in Information and Communication Engineering – XXIX Ciclo

Candidate

Enrico Maiorino
ID number 1195333

Thesis Advisor

Prof. Antonello Rizzi

A thesis submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Information and Communication Engineering

Feb 2016

Thesis not yet defended

Analysis and modeling of complex networks by means of computational intelligence techniques
Ph.D. thesis. Sapienza – University of Rome
ISBN: 000000000-0

This thesis has been typeset by LaTeX and the Sapthesis class.

Version: February 10, 2017

Author's email: enrico.maiorino@gmail.com

Contents

Introduction **v**

1 Complex networks **1**
1.1 Basic properties of networks . . . 1
1.2 Network measures and statistics . . . 3
1.2.1 Centrality measures . . . 3
1.2.2 Assortativity and Modularity . . . 5
1.2.3 Transitivity . . . 8
1.2.4 Shortest paths and small-world effect . . . 8
1.2.5 Degree distribution . . . 9
1.3 Network models . . . 10
1.3.1 Erdos-Renyi graphs . . . 10
1.3.2 Barabasi-Albert graphs . . . 10
1.3.3 Watts-Strogatz graphs . . . 11
1.4 Graph Laplacian . . . 12
1.5 Random walks on graphs . . . 15
1.6 Protein contact networks . . . 16
1.6.1 The Bartoli model . . . 16

2 Heat diffusion on complex networks **19**
2.1 The Considered Networks . . . 20
2.2 Characterization of the Graph Topology . . . 21
2.3 Results . . . 21
2.3.1 Analysis of Topological Descriptors . . . 21
2.3.2 Analysis of the Heat Kernel . . . 22
2.3.3 Canonical Correlation Analysis of the PCA Representations . . . 24
2.3.4 Scaling and Heat Diffusion Analysis . . . 24
2.4 Ensemble Heat Trace . . . 27
2.5 Discussion . . . 30

3 Network as a time series **33**
3.1 Multi-Fractal Detrended Fluctuation Analysis . . . 34
3.2 The Considered Data . . . 36
3.3 Analysis of persistence properties . . . 36
3.3.1 Analysis of multifractal properties . . . 39
3.3.2 Embedding of the multifractal spectra . . . 41
3.4 Discussion . . . 43

4 Detrending of time series with Echo State Networks **45**
4.1 Detrending using ESNs . . . 46
4.1.1 Other detrending methods . . . 48
4.2 Experimental results . . . 49
4.2.1 Synthetic time series . . . 49
4.2.2 Sunspot data . . . 56
4.3 Discussion . . . 59

5 Generation of Protein Contact Networks **61**
5.1 Dataset . . . 62
5.2 First step: the LMGRS generative model . . . 62
5.3 Analysis of the LMGRS ensemble . . . 63
5.4 Second step: the LMGRS-REC reconfiguration procedure . . . 67
5.5 Discussion . . . 69

6 Optimization of the LMGRS networks **71**
6.1 Spectral classes . . . 72
6.1.1 Random matrix theory . . . 72
6.2 Datasets . . . 73
6.3 PCN reconfiguration by means of genetic algorithms . . . 74
6.3.1 Genetic algorithm operators . . . 75
6.4 Results . . . 76
6.5 Discussion . . . 82

A Echo State Networks **85**

Conclusions **87**

Introduction

By observing the world around us, we can identify a variety of environments where a multitude of objects are interacting with each other in complex ways. In many of these situations we can pinpoint some kind of relationship, physical or abstract, between couples of objects. The union of these objects and their interactions constitutes a *network*. Common examples of networks are communities of people, where individuals form relationships with other individuals; power grid networks, where energy is transferred between different geographical locations through power lines; the World Wide Web, probably the most famous kind of network, composed of an enormous number of webpages that are interconnected by hyperlinks.
All these systems can be summarized as a set of *vertices* (or nodes) connected by *edges*. By adopting this representation, we do not always preserve the identity of each vertex or edge, but only their arrangement in the network. The structure that emerges when describing a system in this way is quite minimal. While this may sound like an oversimplification of our description, we obtain an abstraction of the *pure topological structure* of the system we are analyzing. This implies that this description may highlight features about the system that were not evident in more detailed representations. A compelling aspect about this abstract approach is that any conclusion drawn on a purely topological structure such as a network may be generalized to other networks that do not necessarily represent a similar system. The result is the occurrence of common patterns in very different systems that highlight the presence of universal behaviors spanning across different domains. For example, we can find that technological networks such as the internet have similarities with biological networks, or that the growth of a power grid network evolves in a similar way as a financial network.

The field of *complex networks* revolves around this concept of *universality*. The study of phenomena at a global scale is a relatively new approach that is counterposed to the reductionism that has traditionally characterized physical sciences. The famous quote of the physicist and Nobel laureate Philip Warren Anderson, "More is different", is a testament to the importance of searching for fundamental laws at all the layers of abstraction, since often the higher scales of a system can not be described as the combination of laws from the lower scales, and the resulting mechanisms can be inherently different [8]. Not only the total is more than the sum of its parts, the physicist concludes, but it is also different. This is the case for many examples of complex systems. The stock market is composed by a large number of investors whose objective is to maximize their profit, and to a first approximation they could be characterized by very simple rules, yet no one would argue that the resulting network of complicated interactions is simple nor easily described. Complex networks are thus a *simplified representation of a complex system*. There are several key factors that can make a system complex:

- a large number of interacting units;
- nonlinear behaviors or interactions;
- feedback loops in the dynamics;

- phase transitions, i.e. abrupt changes of macroscopic behavior;
- scale invariance;
- chaotic behavior;
- evolution, self-organization and adaptability;
- presence of hierarchical structures;

Not all of these properties are always present, and there is not even an universally agreed-upon definition of the term complex system. However, the main aspect of the research on complex system is its focus on the system's complexity as a topic of study instead of an obstacle to avoid. Given its multi-domain applicability, the analysis of complex systems through their network representation is an interdisciplinary topic that involves the use of methodologies and concepts belonging to very diverse disciplines, like mathematics, physics, computer science, statistics, biology, sociology, economy and so on. A fundamental part of this research involves the analysis of considerable quantities of data and for this reason the recent development of data mining, computational intelligence and machine learning methodologies is a prominent motivation for the rising interest in complex networks.
In this work we aim at employing techniques of computational intelligence and machine learning in the study of complex networks. We analyze a particular kind of networks, namely the Protein Contact Networks, and investigate their peculiarities with an hybrid approach that involves the use of both network-based measures and computational intelligence algorithms. Protein Contact Networks are a representation of the 3D structure of a protein.

A protein is a biological macromolecule that is at the basis of every biological process, like enzyme catalysis, DNA replication, response to stimuli, molecules transport and many others. A protein is composed by one or more long chains of aminoacids residues bonded together by peptide bonds. There are 20 different kinds of aminoacids and the particular sequence of aminoacids that composes a protein is called *primary structure*. The particularity of proteins is the fact that, when they are in solution, they assume a characteristic tridimensional folded shape through a process called *protein folding*[13, 170]. According to the *Anfinsen's dogma*, the 3D structure of the protein in its folded state is completely determined by the primary structure, i.e. by the particular order of aminoacids on the chain. However, the prediction of the folded state of a protein from the information of its primary structure is still an open problem today, referred to as *protein folding problem*. In order to understand why it is important to study the shape of a protein in solution, we must consider that protein are chemical entities with specific functions and responses to stimuli. Specifically, the way for a macromolecule to act on its surrounding environment and to react to external conditions is to change its shape. Indeed, the change of shape affects the interaction potentials between the protein's atoms and the external environment, allowing it to perform a variety of functions. In this regard, a protein can be imagined as a nano-machine equipped with sensors and actuators, and engineered to be as stable as possible from a chemical standpoint. With this parallel in mind it is easy to understand why studying the shape of a protein and how it evolves is tightly related to investigating its function. A better comprehension of how proteins are made and how they work has considerable implications for a variety of fields, in particular medicine. More specifically, proteins are the receptors of nearly all kinds of drugs, and therefore understanding their behavior is of utmost importance in the design of new improved drugs.

However, the large number of atoms and electrostatic potentials at play in a typical protein of even modest dimensions makes the complete analysis—and even the simulation—of such a system quite inaccessible. This inherent difficulty is one of the main reason for the recent

adoption of network-based paradigms such as Protein Contact Networks (PCN) [49]. In the PCN representation, the protein's aminoacids are represented by vertices of a network, and two vertices are connected by an edge if their corresponding aminoacids are in spatial proximity in the tridimensional folded state of the protein, i.e. they are arranged at a distance less than a threshold value. Several choices of the threshold values have been explored in literature, depending on the kind of interactions one wants to include in the analysis, but a very common choice, and the one considered throughout this work, is 8 Å .

This work is structured in two main parts. The first part is an exploratory analysis stage where we analyze and compare PCN with several other kinds of networks and assess their differences and features. The study is carried out with two different approaches. We start by analyzing the properties of the graph-theoretical heat diffusion on the proteins graphs. The diffusion of heat on a graph is a simulated process that is described by the *heat kernel operator* [172] and is deeply influenced by the overall structural organization of the network. Indeed, in tightly connected networks the diffusion of heat is very fast and reaches an equilibrium after a short time. Conversely, a network that is separated in several weakly-communicating modules yields a slower diffusion process. The heat kernel operator provides a series of graph invariants that allow to characterize the topology of a network.
With these graph-theoretic measures we are able to identify a two-regime diffusion process, characterized by a subdiffusive phase for longer times. This behavior is interesting because diffusion of heat on a PCN has an experimental parallel in the case of the protein structure, that is, the energy flow across the residues of a protein. Specifically, a protein molecule is structured in a way such that energy can flow fast through shortcuts connecting distant areas of its structure and otherwise slow along a multitude of pathways reaching dead ends. This behavior is a trade-off that is required for the protein to maintain stability and robustness to random perturbations while still being responsive on a system level to external stimuli. While the existence of this double-regime has been verified experimentally in laboratory studies [100] in this work it has been observed through only graph-theoretic considerations. The analysis stage proceeds with a new approach to the analysis of the structure of a graph. Inspired by the work of Nicosia et al. [134] we investigate long-term correlations properties of random walks performed on protein contact networks in order to infer characteristics of their topology. In particular, this is carried out by studying the multifractal properties of time series composed of observables measured by the random walker on the network's vertices. With this analysis, we evidence the assortative structure of PCN as well as highlight the presence of intra-module and inter-modules link confirming the two-mode spreading of signals mentioned above.
Given the importance of correlation analysis of time series as a proxy for investigation on network structure we also propose a novel detrending method that is able to filter nonstationarity trends in series, in order to avoid the detection of spurious correlations. This method is completely data-driven and is based on the prediction capability of a particular type of neural networks, the Echo State Networks. By means of regularization techniques, we show that Echo State Networks are able to separate trends from statistical fluctuations in data. The resulting detrended series are in turn analyzed for discovering traces of multifractal behavior. By testing this methodology on synthetic dataset we observe state-of-art results with respect to other detrending methods proposed in literature.

Having confirmed the noteworthy peculiarities of PCN, we then proceed with the second part of this project, that is, the generation of new realistic networks that present these characteristics. In other words, the objective of this stage is to design a *generative model* for Protein Contact Networks.
Generative models are a very important tool in network theory. A suitable generative model provides important insights on the evolutionary mechanisms that have lead to the formation of

the system that the network is representing. Moreover, a generative model allows to sample new networks in order to obtain new unseen structures and infer properties that are not easily observable in real data. In the case of PCN, being able to generate new networks is the key to understand the particular trade-offs that evolution has favoured in building such organized and efficient structures. Given the generality of network representation, this knowledge can in turn be transferred to other domains where robustness and efficient transfer of information between distant areas of a network-like system are of fundamental importance in the design of new instances of the system. A prominent example are power grid networks, where flow of energy can be associated to the current flowing along the power lines, and certain trade-offs between stability, economical feasibility and efficiency are required at the design stage.

In the first part dedicated to this topic we propose a generative model of PCN based on a model proposed by Bartoli et al. [20], that we refer to as LMGRS. By performing heat kernel analysis on the generated network, we show that the new generation scheme creates graphs that are more similar to PCN in that they present subdiffusive properties, even if to a lesser extent. However, by analyzing the topological properties of the new networks we observe that they are too tightly connected with respect to their real counterparts. This is probably given by the fact that PCN are the representation of atomic configurations with a non-zero spatial extension, even if microscopic, so not all possible contact between atoms are allowed and, by consequence, not all configuration of the corresponding PCN are possible. To indirectly account for these physical constraints, we propose a reconfiguration scheme that rewires the links in a LMGRS network in order to obtain more realistic configurations. The reconfigured networks, referred to as LMGRS-REC, present statistically significant improvements in similarity with PCN in nearly all the topological properties that we measure. By analyzing the spectral properties of the LMGRS and LMGRS-REC networks we also assess that they show an increased similarity to PCN with respect to the original model of Bartoli et al.

However, since LMGRS networks are still different in terms of spectral characteristics with respect to PCN, in the last part of this work we setup an optimization problem with objective function the spectral similarity between PCN and the candidate solution. More in particular, we aim at obtaining networks whose spectral distribution is as similar as possible as the average PCN spectral distribution. The optimization is performed with a genetic algorithm equipped with custom mutation and crossover operators. These operators are designed in such a way to produce as realistic as possible protein contact networks. The result of the optimization are networks that are indistinguishable from a Protein Contact Network from the points of view of their spectral distribution. This in turn leads to improved similarity of several topological properties.

This thesis is structured as follows: in the first chapter we introduce the field of complex networks and their theoretical background. In the same Chapter, Protein Contact Networks and the Bartoli model are presented. In Chapters 2 and 3 is presented the analysis phase of the networks. In particular, Chapter 2 discusses the heat diffusion properties of PCN and Chapter 3 concerns the random walk analysis on these networks. In Chapter 4 we propose DESN, the data-driven detrending method presented above. Finally, Chapters 5 and 6 discuss the generation of realistic protein contact networks.

Chapter 1

Complex networks

The study of networks has sparked from the recent availability of massive amounts of data allowed by the development of telecommunications, internet in particular, and storage technologies. A network is a very versatile mathematical object that represents the *relation* between different entities. Each entity corresponds to a node (or vertex) of the network and two nodes are connected by an edge if they are related. Examples of networks are:

- social networks, where the individuals are the nodes and edges – in this context often referred to as *ties* – represent their friendship relations with each other;
- the World Wide Web (WWW), where nodes correspond to the web pages connected by their hyperlinks;
- Protein Interaction Networks, describing systems of many proteins and their binary interactions;
- transport networks, where nodes are geographical locations (like cities) and their edges represent aerial, maritime or land transports connecting them.

As it is clear from the examples, the versatility of such objects stems from the fact that nodes and edges can be chosen to represent any kinds of entities and relations, either abstract or physical, and such a choice influences the interpretation of any result obtained on the network. edges can correspond to very different kinds of relations, either abstract or physical.

1.1 Basic properties of networks

A network, also called a *graph* in the mathematical literature, can be formally represented by a list of its nodes and edges. By assigning to each node an arbitrary natural number as unique identifier (ID), each edge is then represented by pairs (i, j) where i and j are the ID of the two connected nodes. Formally, a graph is represented by a pair of sets $G = (\mathscr{V}, \mathscr{E})$, where $\mathscr{V}$ is the set of nodes with $|\mathscr{V}| = N$ and $\mathscr{E} \subseteq \mathscr{V} \times \mathscr{V}$ the set of edges with $|\mathscr{E}| = M$.
An equivalent and often more convenient way to mathematically represent a network is by its *adjacency matrix*. In its simplest form, the adjacency matrix of a graph of n nodes is an $n \times n$ matrix $\boldsymbol{A}$ with elements

$$a_{ij} = \begin{cases} 1, & \text{if nodes (i,j) are connected} \\ 0, & \text{otherwise.} \end{cases} \tag{1.1}$$

The structure of the adjacency matrix completely defines the network. Moreover, since the IDs of the nodes are completely arbitrary, any coordinated permutation of the rows and column of

the adjacency matrix describes the same network. More in general, two graphs having the same structure except for a permutation of their node identifiers are said to be *isomorphic*. A graph is said to be *simple* if it has no self-loops, i.e. nodes connected with themselves, or multi-edges, i.e. multiple edges connecting the same pair of nodes. This in turn corresponds to a zero-diagonal binary adjacency matrix. In the previous examples of networks, the friendship network is represented as a simple graph, since the friendship between two individuals can either exist or not and a self-loop, i.e. a friendship with oneself, would have no meaning in this context. On the other hand, if multiple edges are allowed between pair of nodes, each element a_{ij} can assume any natural value, and the network is said to be a *multigraph*.

When networks edges represent symmetric relations, like in transport networks, the graph is said to be *undirected* and its adjacency matrix is symmetric, i.e. such that $a_{ij} = a_{ji} \; \forall \; i, j \in \mathscr{V}$. In other cases, instead, network edges represent asymmetric relations like, for example, in the WWW network. Indeed, hyperlinks form a directed connection from the source website to the destination website and are not always mutual, i.e. the source site may not be in turn linked back by the destination site. In such a directed graph (also known as DiGraph) the symmetric property of the adjacency matrix does not hold.

Another possible generalization of a graph is possible by allowing the adjacency matrix elements to assume real values. In this case their value represents an "intensity" of connection between nodes and, depending on the context, provides a richer description of their relation. Notice that this value can also be negative, indicating an inhibitory, contrasting or antithetical relation. This kind of network is then said to be *weighted*, since different weights are assigned to its edges.[1] In particular, a multigraph can be interpreted as a specific kind of weighted network where the weights of the edges are constrained to assume natural values. An example of a weighted network is a social network where edges measure a degree of sympathy between the involved individuals. High values corresponds to a strong relationships, negative values to hostility and a null value to neutral or lack of relationship. In some situations it might be necessary or more natural to define a relation where the number of nodes involved is greater than two. *Hypergraphs* are the generalization of graphs that allow to create *hyperedges*, i.e. edges connecting an arbitrary number of nodes simultaneously. Notice that this is not equivalent to creating a fully connected clique of nodes in a conventional graph, since the *group* relation these nodes have in common is lost and only the pairwise connections are retained. As an example, consider a scientific collaboration network, where the nodes are the authors of scientific papers and the edges represent a collaboration in the realization of a paper. In this case the choice of an hypergraph is more natural since there would be a one-to-one correspondence between hyperedges and papers produced. In a conventional graph, instead, for each collaboration of n authors one would be forced to create $n(n-1)/2$ edges between all pairs of authors and the resulting configuration would be indistinguishable from a situation where $n(n-1)/2$ papers are separately produced by the same authors.[2]

The networks defined so far are composed by nodes that are distinguishable only through their wiring patterns. This means that each node has no intrinsic identity and different nodes can be swapped without altering the network's properties. In situations where the identities of nodes are not interchangeable it is possible to assign a label to each node, obtaining a *labeled* graph. A label is any kind of data associated to the entity the node represent, and is considered when

[1] In many situations it is also useful to extract and analyze the purely topological substrate of a weighted network, that is, the unweighted network obtained by removing all the weights from its edges. The obtained structure can highlight features that are not immediately obvious in the original network and allows for the application of measures and procedures developed for unweighted networks.

[2] The downside of choosing the hypergraph notation is that since hyperedges are sets instead of 2-tuples they are harder to handle and many methods available in literature are not generalizable to this kind of objects.

comparing graphs or in subsequent processing[3].
A summary of all the possible generalizations of a graph is shown in Table 1.1. In this thesis we will be focusing only on simple graphs, given their ease of description and diffusion in the complex networks literature. Additionally, we will be considering connected networks, i.e. networks such that any pair of nodes is connected by a path across the edges.

1.2 Network measures and statistics

One of the most common approaches to study a network and highlight its main organizing principles is to search for suitable *measures* and *statistics* that reduce a local or a global topological property to a numeric value or distribution. The obtained values can then be used for comparisons with other graphs or to get insights on hidden wiring patterns within the network. Measures and statistics have been defined in the complex network literature in order to highlight different features of the network, like e.g. discover central nodes belonging to the network's "core", predict the presence of edges, identify clusters of tightly-connected nodes, etc. In this section we describe several common measures and statistics on simple graphs that are evaluated and discussed in subsequent sections and are commonly employed in network analysis.

1.2.1 Centrality measures

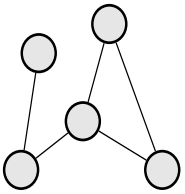
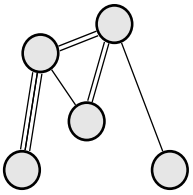
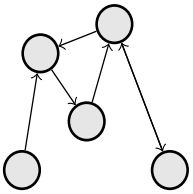
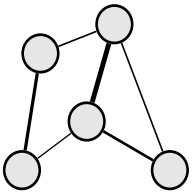
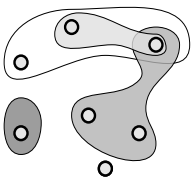
A lot of real-world networks are composed by a great number of interconnected nodes. It is in general unlikely that all the nodes have the same importance and hierarchical role in the organization of the network. A common example is the world airline network, modeling the world airports as nodes and the direct air routes connecting them as the edges of the graph. In such a network it is easy to recognize the important role that large international airports have in the organization of traffic across the network. These airports are generally connected with a great number of smaller airports, so the correct functioning of these airports is crucial for the connection of many faraway locations. In this sense, these airports constitute the "central core" of the network, while smaller airports are distributed across the "periphery". The most intuitive way to quantify the centrality of a node in a network is to consider its *degree*, defined as the sum of its connections with other nodes. By considering a simple graph of N nodes with adjacency matrix $A = \{a_{ij}\}$, the degree k_i of node i is

$$k_i \equiv \deg(i) = \sum_{j=1}^{N} a_{ij} \tag{1.2}$$

Despite its simplicity, the *degree centrality* is in many cases an accurate descriptor of the importance of the node in a network. For example in a social network where nodes are individuals and edges are their relationships it is reasonable to assume that nodes with a large degree, often referred to as *hubs*, correspond to the most influential individuals and are critical in the study of information spreading, group dynamics, etc. The concept of importance, however, is obviously dependent on the particular system that the network is modeling and the hypothesis to be investigated. In some cases one needs a measure of the centrality that describes how well-connected is a node to the rest of the network. In the previous world airline network example, one could identify as central those airports which are at the least number of steps away from any other location in the network. This means that, starting from a central node, any

[3]Clearly, the label does not affect the pure topology of the graph, even if it can be taken into account when defining custom measures.

Table 1.1. Summary of the main variants of a graph. The fields marked with "-" correspond to properties that are not required to be fixed for the graph definition. For example, the adjacency matrix of a directed graph can be either binary, natural or real. Graph definitions can also be combined by joining their respective properties, like e.g. a directed multigraph is a graph with multiple edges and a non-symmetric adjacency matrix. In the last example graph, hyperedges are represented by Venn diagrams and nodes are drawn as black dots.

Graph type	**Edge type**	**Elements a_{ij}**	**Adj. matrix type**	**Example**
Simple graph	2-tuple	Binary	Zero-diagonal	
Multigraph	2-tuple	Natural numbers	-	
Directed graph	2-tuple	-	Non-symmetric	
Weighted graph	2-tuple	Real	-	
Hypergraph	Set	Undefined	-	

other node of the network can be reached through a small number of 'hops' across the edges of the graph. The corresponding measure, called *closeness centrality*, should assign high values to such nodes and low values to remote nodes. While several definitions of this measure have been proposed, one of the most common, and the one used in this thesis, has the form

$$C_i = \frac{N}{\sum_j d_{ij}} \tag{1.3}$$

where d_{ij} is the length of a *geodesic path* between nodes i and j, i.e. the shortest path between i and j with each edge having unitary length. While this measure is usually correlated with degree centrality, it is a property that depends on the whole structure of the graph, so it provides in principle a different kind of information about the position of the node in the network [163]. In other situations one could consider a completely different criterion to quantify centrality in the network. Consider, for example, a power grid network, where nodes represent power stations and edges their physical connections. When dimensioning the capacity of a power station one has to account for the maximum electrical flow it can sustain. In a first approximation, the station will handle an electrical flow that is directly proportional to the number of shortest paths traversing its corresponding node in the network. In this definition of centrality, the more the station serves as a "bridge" between the other endpoints of the network, the more the node is central. Conversely, peripheral nodes are those sites that are not fundamental in the connection of different areas, like end-users houses in the power grid example. This kind of centrality is defined as *betweeness centrality* and is a fundamental measure to consider when discussing the efficiency of transport/transmission of a node in a network and the consequences of its malfunctioning. The betweeness centrality $C_i^{(B)}$ of a node k is calculated as

$$C_k^{(B)} = \frac{\sum_{ij} \sigma(i,j|k)}{\sum_{ij} \sigma(i,j)} \tag{1.4}$$

where $\sigma(i,j)$ is the number of shortest paths between nodes i and j and $\sigma(i,j|k)$ is the number of shortest paths between i and j passing through node k.[4] While this property can be related to the degree centrality, there are cases where this connection does not hold. For example, consider the network in Fig. 1.1: the node highlighted in red has a degree of 2, yet it has high betweeness centrality since it is the only connection between the two subgraphs shown on its left and right. As a consequence, the removal of this node would have dramatic consequences for the global topology since the network would become disconnected.

While the centrality measures defined above are the only types discussed throughout this work, the list is certainly not complete and new centrality measures are constantly being defined in literature. This derives from the fact that centrality in a network is a strongly *application-dependent* concept and the incorrect application of these measures can lead to a wrong interpretation of data.

1.2.2 Assortativity and Modularity

Networks are a very minimalistic way of representing a system. For a system composed of many entities this corresponds to focusing on their relations more than their individual identities. One of the main questions that could arise when analyzing a network is: "Why are nodes connected in this way? Is there some sort of wiring principle behind the resulting

[4] There can be multiple paths with the same length connecting two nodes, so while the shortest path length is always univocally determined, the shortest path itself is not.

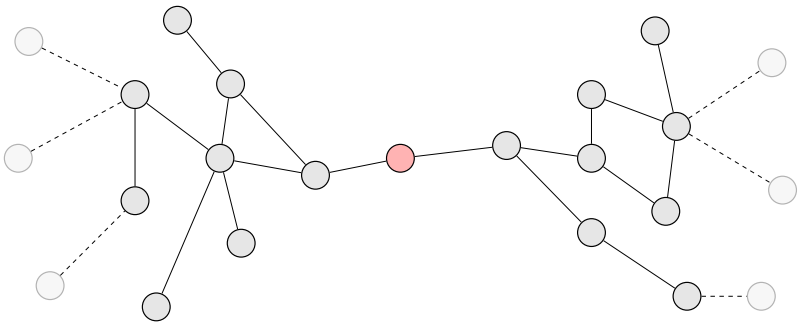


Figure 1.1. Test

configuration?". Identifying the properties that affect the connection between two nodes can give an important insight on the system represented by a network. In many real-world networks a first step in this direction is to analyze the existence of *homophily* between nodes. Homophily, or *assortative mixing*, corresponds to the tendency of the nodes to be connected with other nodes with similar properties. This is very common for example in social networks, where individuals tend to be connected with people that are similar in some relevant characteristic, like social class, political orientation or geographical location. Conversely, there are networks where the opposite is more likely, and a *disassortative mixing* is observed, i.e. nodes with very different characteristics are more likely to be connected. The *assortativity* of a network is the statistic that quantifies this tendency. Let us consider a connected network with N nodes and M edges and adjacency matrix A_{ij}, and a nominal observable $O(i) \equiv O_i$ that can be evaluated on any node i. The property O is nominal in the sense that it assumes symbolic values, i.e. non-numeric values that have no intrinsic ordering nor product, like political orientation in the example above[5]. In its original form, the assortativity of the network with respect to the observable O is calculated by counting the fraction of edges connecting nodes with the same value of the property O, and by subtracting the contribute given by the probability of them being connected by pure chance. The assortativity has the form

$$Q = \frac{1}{2M} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2M} \right) \delta(O_i, O_j) \tag{1.5}$$

where k_i stands for the degree of node i and $\delta(O_i, O_j) = 1$ if $O_i = O_j$, and zero otherwise. Let us now analyze the form of the previous equation. The fraction of edges neighboring nodes with the same value of O is given by the first term of eq. (1.5), i.e.

$$\frac{1}{2M} \sum_{ij} A_{ij} \delta(O_i, O_j).$$

This value alone would not keep into account the connections given by pure chance and not because of an assortative relations between the nodes: consider for example nodes with a very high degree k. These nodes are more likely to be connected with each other in virtue of their higher number of connections, even not taking into account the value of their observable O. On the other hand, consider a network with a very unbalanced distribution of values of the observable O on the nodes, so that many nodes have the same property O^*. In this case the

[5]Even if a numerical label can be assigned to each symbol, the resulting numerical ordering would be arbitrary and hence meaningless. This corresponds to stating that equality is the only well-defined logic operation between the values of the domain of O, while the inequalities are not.

fraction of nodes with the same value of the observable O would be very high even when choosing the neighbors of each node in a random fashion. From this considerations it is clear that a useful measure has to highlight the presence of some interesting pattern while ignoring coincidental properties given by pure chance or forced association (like in a graph where all the observables have the same value). In the statistics language, it means that the measure has to provide some comparison to a suitable *null model*. Null models are an important tool in network science (and statistics in general) because they allow to compare the observed data to a model where correlations have been removed in a controlled fashion, by randomizing existing data. This allows to attribute any observed deviation of data from the null model as the result of correlations that are not present in the null model. On the other hand, any effect observed both in the null model and data can then be attributed as the result of randomness. Depending on the particular feature that one wants to investigate it is possible to design a randomization scheme whose realizations retain the properties to be considered as fixed (and hence not interesting for the analysis) and the rest are randomized. Consider a random graph with the same N nodes as the original graph but where each node i is connected with k_i chosen at random, where k_i is the node's degree in the original graph. The set of possible realizations of this random graph defines a *null model* of the original graph, where the degree of each node is conserved but the connections between nodes are not, the so-called *configuration model*. In a configuration model with the same degree distribution as the original graph, the probability of two nodes with degrees k_i, k_j to be connected is $\frac{k_i k_j}{2M}$. From this viewpoint, the term in the parenthesis in eq. (1.5) represents the difference between the actual value of the adjacency matrix and its expected value if nodes were wired in a random fashion. In this way, the assortativity in its standard form compares the wiring patterns of the graph to its corresponding configuration model for some observable O. Assortativity is bounded in the range $[-1, 1]$ and it assumes positive values when the graph is assortative, negative values when it is disassortative and it is zero in absence of correlations.

When the observable O corresponds to labels identifying different communities of nodes, like for example the nationality in a social network, the assortativity corresponds to the *modularity* of the graph [27, 133]. Modularity is a measure of how clustered is the graph with respect to the labeling given by the observable O. In particular, in a modular graph nodes are more likely to be connected with nodes of the same community than with other nodes. Considering the nationality example, strong modularity is often observed in social networks with respect to nationality since people sharing a similar cultural background are more likely to interact and form relationships.

When the considered observable is a scalar value, it is convenient to define the *assortativity coefficient* as the generalization of the Pearson correlation coefficient over graphs:

$$r = \frac{\sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2M} \right) O_i O_j}{\sum_{ij} \left(k_i \delta_{ij} - \frac{k_i k_j}{2M} \right) O_i O_j}. \tag{1.6}$$

The term at the numerator corresponds to a covariance term and the numerator to a variance term. The assortativity coefficient gives an idea on how correlated are the observables on neighboring nodes. A very common choice of node observable that turns out to be surprisingly informative is the degree of the node. The degree can be always calculated, since it doesn't depend on external information, and for this reason provides insights on the pure topologic structure of the network. In particular, the *degree assortativity coefficient* measures the tendency of nodes of being linked to nodes of similar degree. If the network is assortative with respect to the degree, then it is usually composed of a main core of high-degree nodes, or hubs, tightly

connected with each other and a periphery of many interconnected low-degree nodes[6]. An example of assortative networks are the film actor networks, that is, networks where each node represents an actor and edges represent the starring of two actors in the same movie. In this kind of networks the degree is correlated with the celebrity status of the actor, and for this reason it is likely that actors with high celebrity statuses collaborate in high-budget movies. The opposite of a degree assortative network is a network where hubs are preferentially connected to low degree nodes. This situations happens, for example, in technological networks where hubs represent servers and low-degree nodes are the clients that request a service.

1.2.3 Transitivity

The assortativity measure presented above is a way to highlight the tendency of pairs of nodes to be connected. Another important information that we can extract from the local wiring of a group of nodes is the transitivity property of their connections. Transitivity in this context means the tendency of a node to be connected with the neighbors of its neighbors. In other words, if a node i is tied to a node j and j is tied to a node k, in a transitive network there is a high probability that i is in turn tied to node k. Transitivity is a common feature of many real-world networks, like in friendship networks, where if an individual i is friend with an individual j and j is friend with k then it is more likely that i is friend with k than with some random node in the network. One of the ways to quantify transitivity with in a graph is the *clustering coefficient*. With respect to a single node i of degree k_i the clustering coefficient c_i is defined as

$$c_i = \frac{2N_{\mathrm{tr}}(i)}{k_i(k_i-1)}, \tag{1.7}$$

where $N_{\mathrm{tr}}(i)$ is the number of pairs of neighbors of i that are connected. Since the denominator corresponds to the total number of pairs of neighbors of i (regardless of their connection), the measure in eq. (1.7) is a fraction of the transitive relations involving node i with respect to the total. The coefficient is hence bounded between 0 and 1. The value 0 corresponds to the absence of transitivity, while the value 1 denotes that i and its neighboring nodes form a *clique*, i.e. a completely connected subgraph. The global clustering coefficient of a graph is then the average of all the clustering coefficients of its nodes, i.e.

$$C = \frac{1}{N}\sum_i c_i. \tag{1.8}$$

While a global clustering coefficient of 1 corresponds exclusively to a completely connected graph, there are a multitude of topologies that correspond to a null clustering coefficient, like trees, lattices with degree greater than 3, etc.

1.2.4 Shortest paths and small-world effect

As already mentioned in Sec. 1.2.1, an important feature of a graph topology is the distribution of topological shortest paths between nodes. For example, when traveling by plane from a location to another, one tends to chose the path with the minimum number of stopovers more than the path with the shortest geographical distance. A well-connected network will then be characterized by short paths between pairs of nodes, so as to minimize the number of steps

[6] Even in assortative networks hubs will be necessarily connected with many low-degree nodes, but notice that the assortativity coefficient compares their actual frequency of connection with the corresponding expected probability in a random model, so any forced connection is ruled out.

necessary to go from a node to any other. One way to quantify this connectedness property of a network is by evaluating its *diameter*, defined as

$$D = \max_{i,j} d_{ij} \tag{1.9}$$

with d_{ij} the shortest-path length between nodes i and j. The diameter measures the worst-case path length in the whole network. However, in many cases this measure turns out to be too sensitive to few outliers since it only depends on the longest path between nodes and not on the global distribution of paths. A more meaningful measure is the *average shortest path*, calculated by substituting the maximization in eq. (1.9) with an average:

$$\tilde{D} = \frac{1}{N(N-1)} \sum_{i,j} d_{ij} \tag{1.10}$$

This quantity is more robust to outliers and is dependent of the whole distribution of paths, even if some care has to be taken in case of very heterogeneous distributions of path lengths. It has been observed that in many real-world networks the average shortest path grows very slowly with the network size, that is, even large networks tend to have very small average shortest paths. Such networks are said to be *small-world*, in virtue of their apparent small size because of their efficient connectivity. The most famous example of this effect is the anecdote of the “six degrees of separation", the conjecture that any two persons in the world are separated by only six steps in terms of friendship relations with other people. This conjecture originated by several experiments conducted by the social psychologist Stanley Milgram, that observed that many social networks are indeed small-world. Formally, a network is said to be small-world when its average shortest path grows logarithmically or slower with the number of nodes. However, it can be shown that the small-world property is not necessarily an hint of some underlying process optimizing the network’s connectivity, since it can be shown that several models of random graphs yield a small-world topology.

1.2.5 Degree distribution

As discussed in Sec. 1.2.1, the degree is an important feature of a node since it gives a general understanding of the node connectivity with the rest of the network. On the other hand, in many situations it is important to identify the distribution of degrees at a network level, regardless of the identity of the specific vertices, in order to identify the balance between high-, medium- and low-degree nodes. This in turn provides information about the structural organization of the graph, as well as many other insights on its robustness, redundance, efficiency of connection, etc. depending on the specific system it is representing. The degree distribution $P(k)$ is the probability distribution of measuring a given degree k by randomly selecting a node of the network. This probability can be easily estimated by evaluating the empirical frequency of each degree k. By drawing an histogram of the resulting data it is possible to observe the general trend of $P(k)$ as a function of k. In most real-world networks this trend is decreasing since generally any meaningful network representation of a system is sparse in its edges and high-degree nodes are rarer than low-degree ones. By representing the histogram in a loglog plot, in many situations the resulting probability distribution of degrees turns out to have the form

$$\log P(k) = -\alpha \log k + c. \tag{1.11}$$

This form is recognizable by observing an approximately linear trend in the loglog plot. The form in eq. (1.11) corresponds to a power law function, indeed

$$P(k) \propto k^{-\alpha}. \tag{1.12}$$

The α coefficient represents the decay of the power law as a function of the degree: lower values of α lead to the presence of very large hubs with degrees that are significantly higher than the mean degree. Networks whose distribution is a power law are said to be *scale free*, since the degree of their nodes have no characteristic scale. In this kind of networks the average degree value loses much of its significance since the variance of the resulting distribution is very large, or even infinite. In the next Section we discuss the prototypical model of a scale free network, the Barabasi-Albert model.

1.3 Network models

One of the best ways to recognize an important pattern within a real-world network is to design a suitable null-model with minimal assumptions that produces some effects on data. This topic has been mentioned in Sec. 1.2.2, regarding the definition of a meaningful assortativity coefficient. For this reason in the network literature many generative models have been proposed aiming at producing these interesting patterns and discover when some effect is not a result of random fluctuations. In this Section we briefly describe some of the most famous network models that are commonly used in literature because of their analytical simplicity.

1.3.1 Erdos-Renyi graphs

The Erdos-Renyi (ER) graphs are the most basic kind of random graphs. An ER network is generated starting from N disconnected nodes. For each pair of distinct nodes a link is created with a fixed probability p, so that each link is uncorrelated with each other. Despite their simplicity, ER graphs are a very important aspect of network theory since they are among the few models that can be studied analitically and show many non-trivial features that can be calculated exactly. In particular, it can be shown that for suitable values of p ER graphs are indeed small-world. ER graphs have also been thoroughly analyzed from many other viewpoints, like clustering, percolation properties, resilience, degree distribution, etc. [5].

1.3.2 Barabasi-Albert graphs

Many systems are characterized by a steady growth process where new individual entities are gradually created or added and these interact with existing components of the system. The topology of the graph at any given time is then the result of a dynamical process. For example, consider the World Wide Web network, where each node is a site domain and hyperlinks between sites are the edges. In the early days of internet this network would be composed by few thousands of nodes. Gradually, new domains have been registered and new links are created in order to obtain the actual WWW structure. The Barabasi-Albert (BA) model is a very simple schematization of this kind of growth process. The main principle behind the growth is the *preferential attachment* mechanism, according to which new nodes that join the network tend to connect to existing nodes with a probability that is proportional to the degree of these nodes. In particular, if a node has a very high degree, then it has more probability of being connected to the newly created nodes, which in turn causes its degree to increase even more. Conversely, nodes with low-degree at any given time are less likely to gain new links and hence their relative degree decreases further with respect to the most connected nodes. This leads to a very wide distribution of degrees and to the so-called "rich get richer" effect, which is a behavior that has been observed in many domains like sociology, biology, finance, and so on. The steps to construct a BA graph are as follows:

1. start from a complete graph with M connected nodes;

2. create a new candidate node with degree M and assign each endpoint of its links to an existing node selected with a probability p_i, given by

$$p_i = \frac{k_i}{\sum_j k_j} \tag{1.13}$$

where k_i is the degree of the i-th node and the sum in the denominator is over all the existing nodes except the candidate node[7];

3. add the candidate node to the network and repeat from point 2 until a given network size of N has been reached.

It can be shown [5] that for $N \to \infty$ the degree distribution of a BA graph has the form

$$P(k) \propto k^{-\gamma} \tag{1.14}$$

where $\gamma = 3$ is a decay exponent that is independent of M. As discussed in Sec. 1.2.5 this form of the degree distribution stems from a scale free topology. Indeed, BA graphs are characterized by a small number of very large hubs and a large number of low-degree nodes.

1.3.3 Watts-Strogatz graphs

In Sec. 1.3.1 it was mentioned that ER graphs are characterized by a small world topology. However, in the early years of research on the small-world property it was observed that ER graphs had a major discrepancy with small-world networks extracted from real data. Indeed, while the average shortest path was comparable, real networks showed much higher clustering coefficients with respect to ER graphs. In Ref. [168] this was attributed to some kind of local order between vertices that was lost in random graphs. In the same work the authors proposed a new generative model of small-world networks that showed high clustering while still retaining a small average shortest path. This model is structured as follows:

1. start from a connected regular ring lattice of N nodes, where each node is linked to its k nearest neighbors where N and k can be of any values;[8].

2. choose a node and the edge that connects it with its nearest neighbor, and with probability p we rewire the edge to any other node of the network, avoiding any multiple edges, and with probability $1 - p$ we leave the edge in place;

3. select the next node by moving clockwise on the ring and perform the same operation as in step 2 until we return to the first selected node;

4. when the first node is selected again we perform the same operation as in step 2 with its the second nearest neighbor on the lattice and move clockwise until all edges of the second nearest neighbors have been considered;

5. we perform the same operation as in step 4 for third, forth, etc., nearest neighbors until all edges have been considered

The free parameters for this model are the number of nodes N, the rewiring probability p and the number of neighbors k. By varying p in the range $[0, 1]$ one obtains varying levels of randomness, starting from a regular ring lattice for $p = 0$ and ending up to a ER graph for $p = 1$.

[7] In order to obtain a simple graph we ignore nodes that are selected twice, or reassign the link to a new node. However, this situation is negligible when the network grows to a size $N \gg M$

[8] Usually for k is chosen an even number for symmetry

1.4 Graph Laplacian

An alternative representation of a network that can be computed directly from its adjacency matrix is the graph Laplacian, that is strongly related to the diffusion properties of the graph. Consider a graph with adjacency matrix $\boldsymbol{A}$. We can then define the degree matrix $\boldsymbol{D}$ as the diagonal matrix with element $D_{ij} = k_i \delta_{ij}$. The Laplacian matrix is then defined as

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}. \tag{1.15}$$

and each element is equal to

$$L_{ij} = \begin{cases} 1 & \text{if } i = j, \\ -1 & \text{if } A_{ij} = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{1.16}$$

The Laplacian $\boldsymbol{L}$ is the generalization of the differential Laplacian operator to graphs [92] and provides important information on the topology of the graph and is fundamental in many contexts of network theory like graph partitioning, dynamical processes on the network, network connectivity, etc [46, 97, 138, 176]. Moreover, the Laplacian is directly related to many local and global properties of the network [120, 124], and its spectrum provides a compact, one-dimensional representation of the graph which can be probed for understanding the founding principles behind the network's organization [34, 122, 175, 178]. To understand why $\boldsymbol{L}$ has the form in eq. (1.15), consider a graph with N nodes and degrees $\{k_i\}$ and consider each node as a container filled with an hypothetical substance that diffuses over the graph's edges. This substance may represent some physical quantity like a density of gas, quantity of cars in a road transport network, etc. or even an abstract quantities like information, influence and so on. At the starting time there is some initial quantity of substance ψ_i on each node i, and since at any instant it propagates to the neighboring nodes by diffusion over the edges, its evolution is described by the differential equation

$$\frac{\partial \psi_i}{\partial t} = C \sum_j A_{ij} (\psi_j - \psi_i) \tag{1.17}$$

where C is a characteristic diffusion constant. The right-hand side of previous equation can be rearranged so as to obtain

$$\frac{\partial \psi_i}{\partial t} = C \sum_j (A_{ij} \psi_j - \delta_{ij} k_i) \psi_j \tag{1.18}$$

which in vector form equals to

$$\frac{\partial \boldsymbol{\psi}}{\partial t} = C(\boldsymbol{A} - \boldsymbol{D}) \boldsymbol{\psi} \tag{1.19}$$

from whence it derives

$$\frac{\partial \boldsymbol{\psi}}{\partial t} + C \boldsymbol{L} \boldsymbol{\psi} = 0 \tag{1.20}$$

which has the same form as a diffusion equation except for the differential Laplacian operator $\boldsymbol{\nabla}^2$ that has been substituted by the graph Laplacian $\boldsymbol{L}$.

Starting from the graph Laplacian it is possible to define the *normalized graph Laplacian* $\tilde{\boldsymbol{L}}$ as

$$\tilde{\boldsymbol{L}} \equiv \boldsymbol{D}^{-1/2} \boldsymbol{L} \boldsymbol{D}^{-1/2} = \boldsymbol{I} - \boldsymbol{D}^{-1/2} \boldsymbol{A} \boldsymbol{D}^{-1/2} \tag{1.21}$$

and each elements is equal to

$$\tilde{L}_{ij} = \begin{cases} 1 & \text{if } i = j, \\ -\frac{1}{\sqrt{d_i d_j}} & \text{if } A_{ij} = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{1.22}$$

The normalized graph Laplacian is an important, formal representation of graphs, since it conveys many structural and dynamical properties of the modeled system [107, 117, 122]. Moreover, the normalized laplacian is generally studied because of its interesting spectral properties, as shown and discussed throughout this work. In particular, the normalized graph Laplacian can be decomposed according to the spectral representation

$$\tilde{\boldsymbol{L}} = \boldsymbol{\Phi}\boldsymbol{\Lambda}\boldsymbol{\Phi}^T, \tag{1.23}$$

where

$$\boldsymbol{\Lambda} = \text{diag}(\lambda_1, \lambda_2, ..., \lambda_N) \tag{1.24}$$

is a diagonal matrix containing the eigenvalues, $\boldsymbol{\Phi}$ is the eigenvectors matrix. It can be shown that the eigenvalues of the *normalized Laplacian spectrum* are bounded between 0 and 2 and the smallest eigenvalue is always 0, so we can choose the to order the indices such that $0 = \lambda_1 \leqslant \lambda_2 \leqslant \lambda_3 \leqslant \cdots \leqslant \lambda_N \leqslant 2$. One interesting property of the normalized spectrum is that the multiplicity of the eigenvalue 0 is the equal to the number of connected components of the graph (hence the reason why there is always at least one zero eigenvalue). Moreover, the highest eigenvalue is always lesser than 2 expect for bipartite graphs, where it is 2. The influence of the normalized laplacian spectrum on the graph's topology is one of the main focuses of this thesis and it is thoroughly analyzed with several approaches.

Starting from the normalized Laplacian, it is possible to define a diffusion equation similar to eq. (1.20), called *heat equation* [96, 172]. The heat equation has the form

$$\frac{\partial \boldsymbol{H}_t}{\partial t} = -\tilde{\boldsymbol{L}}\boldsymbol{H}_t, \tag{1.25}$$

where $\mathbf{H}_t$ is a time-varying doubly-stochastic $n \times n$ matrix, called *heat kernel*, and t is the time variable. The heat equation describes the flow of heat over the graph and, differently from the diffusion equation of eq. (1.20), here the unknown variable is a matrix instead of a vector. Indeed, the heat equation describes the flow of some diffusing substance over the network edges rather than the quantity of substance accumulated in the nodes. This is why the heat equation is particularly suitable to describe the flux of information from a vertex to another in situations where cascade interactions occur. The general solution to (1.25) is

$$\boldsymbol{H}_t = \exp(-\tilde{\boldsymbol{L}}t), \tag{1.26}$$

which can be calculated by exponentiating the spectrum of $\tilde{\boldsymbol{L}}$:

$$\mathbf{H}_t = \boldsymbol{\Phi}\exp(-\boldsymbol{\Lambda}t)\boldsymbol{\Phi}^T = \sum_{i=1}^{N} \exp(-\lambda_i t)\boldsymbol{\phi}_i\boldsymbol{\phi}_i^T. \tag{1.27}$$

The solution $\boldsymbol{H}_t$ of eq. (1.25) is a time-varying matrix whose $(i,j) - th$ element represents the quantity of substance that originated from node i and is flowing towards j by following a path with a length of t. Notice that for this reason in a connected graph the heat matrix is in general non-zero everywhere for a sufficiently high time instant t. In particular, $\boldsymbol{H}_t \simeq \boldsymbol{I} - \tilde{\boldsymbol{L}}t$ for $t \to 0$ and

$\boldsymbol{H}_t \simeq \exp(-\lambda_2 t)\boldsymbol{\phi}_2\boldsymbol{\phi}_2^T$ when t is large[9]. This means that the large-time behavior of the diffusion depends on the global structure of the graph, while its short-time characteristics are determined by the local structure. The *heat trace* $\mathrm{HT}(t)$ of $\boldsymbol{H}_t$ is given by

$$\mathrm{HT}(t) = \mathrm{Tr}(\mathbf{H}_t) = \sum_{i=1}^{N} \exp(-\lambda_i t) = C + \sum_{i=C+1}^{N} \exp(-\lambda_i t), \tag{1.28}$$

where C is the multiplicity of the eigenvalue 0, corresponding to the number of connected components of the graph. For a connected graph the heat trace is hence

$$\mathrm{HT}(t) = 1 + \sum_{i=2}^{N} \exp(-\lambda_i t). \tag{1.29}$$

The heat trace assumes the value N at $t = 0$, where N is the number of nodes, and approaches asimptotically the value 1 for $t \to \infty$. To obtain a clearer interpretation of the heat trace, we consider a scenario where the (graph-theoretical) heat represents the information transfer between nodes. Each node generates as time $t = 0$ a message that is propagated across the network edges. The value $\mathrm{HT}(t)$ can then be interpreted as the number of separated components of the graph at the time t from the point of view of information exchange. Indeed, at time 0 the graph is equivalent to N separated components since no communication has been established between nodes. When time t grows the information generated on each node reaches farther and farther locations of the network. For $t \to \infty$ an equilibrium state is reached where each node is in contact with any other node of the network and there is no more information exchange. In this perfectly synchronized state the network is a single connected component from the information viewpoint and this is reflected in the unitary value of the heat trace. An interesting property of the heat trace is that its value is invariant under permutations of the node ordering, so isomorphic graphs yield identical curves. On the other hand, the heat trace is a scalar quantity that depends only on the eigenvalues of the normalized Laplacian, so it does not preserve the information about its eigenvectors. A related quantity that depends both on eigenvalues and eigenvectors is the *heat content*. The heat content $\mathrm{HC}(t)$ of $\mathbf{H}_t$ is defined as:

$$\mathrm{HC}(t) = \sum_{u\in\mathcal{V}} \sum_{u\in\mathcal{V}} \mathbf{H}_t(u,v) = \sum_{u\in\mathcal{V}} \sum_{u\in\mathcal{V}} \sum_{i=1}^{N} \exp(-\lambda_i t)\phi_i(v)\phi_i(u). \tag{1.30}$$

It can be shown (see Ref. [116]) that the heat content can be expressed in terms of power series expansion,

$$\mathrm{HC}(t) = \sum_{m=0}^{\infty} q_m t^m. \tag{1.31}$$

By using the McLaurin series for the exponential function, we have

$$\exp(-\lambda_i t) = \sum_{m=0}^{\infty} \frac{(-\lambda_i)^m t^m}{m!}, \tag{1.32}$$

which, substituted in Eq. 1.30, gives

$$\mathrm{HC}(t) = \sum_{u\in\mathcal{V}} \sum_{u\in\mathcal{V}} \sum_{i=1}^{N} \exp(-\lambda_i t)\phi_i(v)\phi_i(u) = \sum_{m=0}^{\infty} \sum_{u\in\mathcal{V}} \sum_{u\in\mathcal{V}} \sum_{i=1}^{N} \phi_i(v)\phi_i(u)\frac{(-\lambda_i)^m t^m}{m!} \tag{1.33}$$

[9]The vector $\boldsymbol{\phi}_2$ is referred to as *normalized Fiedler vector*

and thus

$$q_m = \sum_{i=1}^{N} \left(\sum_{u \in \mathscr{V}} \phi_i(u) \right)^2 \frac{(-\lambda_i)^m}{m!}. \tag{1.34}$$

The q_m coefficients in Eq. (1.31) are called *heat content invariants* (HCI) and have been shown to be quite informative of the global network structure [172].

1.5 Random walks on graphs

Networks in their graph representation are generally awkward to handle from a mathematical viewpoint. Their structure is invariant under permutations of the node ordering, and they are in general infinite-dimensional objects[10]. For this reason in literature many methods have been defined to linearize the structure of a graph and obtain some 1-dimensional description of its structure. In Ref. [134] the authors propose to design a random walk process on the network in order to probe its structural properties. In particular, let us consider a graph G and a random walker that at the time step $t = 0$ is randomly placed onto a node of the graph. The walker evaluates some kind of property of the node, for example the degree, and stores the value in a list. At $t = 1$ the walker jumps on a node chosen randomly among the previous node's neighbors, then registers the value of the measured observable on the new node and adds it to the list. The process goes on for a T iterations and the output is the time series of T values of the observable measured during the random walk across the graph's edges. This process is Markovian and its evolution can be described by evaluating the *transition matrix* associated to the graph's adjacency matrix A, given by

$$\boldsymbol{P} = \boldsymbol{D}^{-1}\boldsymbol{A}. \tag{1.35}$$

where $\boldsymbol{D}$ is the degree matrix. More precisely, the matrix $\mathbf{P}$ defines a first-order, unbiased Markovian chain on the nodes of G, where the transition probabilities are entirely determined by the adjacency matrix [30, 33]. If G is connected and undirected, then the underlying Markov process admits a unique *stationary distribution* on the nodes, $\boldsymbol{\pi} = \boldsymbol{\pi}\boldsymbol{P}$, given by the vertex degrees:

$$\pi_i = \frac{k_i}{2M}. \tag{1.36}$$

We now define a time-homogeneous property map $M : \mathscr{V} \to \mathscr{O}$, where $\mathscr{O}$ is the domain of vertex properties, such as degree, clustering coefficient or other well-defined observables of the nodes of G. A random walk on a graph G generates a sequence of vertices that are visited over discrete time indexes. We can associate to the random walk a sequence of observables by evaluating at each time step t the corresponding vertex observable given by $O_t = M(v_t)$, where v_t is the node being visited at time t, generating thus a sequence of nodes properties, $O_1, O_2, ..., O_T$. Interestingly, the process described above is technically equivalent to an hidden Markov model where the observables are determined deterministically from the current state. As shown in

[10] To see why, consider a regular square lattice where each node has exactly $2D$ neighbors. This network can be embedded in a D-dimensional Euclidean space since the symmetry and triangular inequality of the Euclidean metric are respected by definition. Indeed, in this space there is a local order where each node cannot be linked to any other node of the network. Consider the previous example, and suppose to create a link between two distant nodes on the lattice. In the D-dimensional space, this 'shortcut' violates the triangular inequality and so the graph has to be embedded in a higher-dimensional space to restore it. By allowing an arbitrary number of nodes and shortcuts between arbitrarily distant areas of the lattice one obtains a general topology and the embedding space dimension grows accordingly, becoming in general infinite. The interested reader may see Ref. [32] for further discussion on the topic.

Ref. [134] and in Chapter 3, by studying the correlation properties of the resulting time series it is possible to obtain surprisingly informative insights on the structural organization of the network.

1.6 Protein contact networks

As we will discuss in the following Chapters, the main object of study of this work are *Protein Contact Networks*. Protein contact networks are a graph theoretic representation of a protein 3D structure in the folded state. Aminoacids are represented as vertices of an unweighted graph, and two vertices are connected by a link if the corresponding aminoacids are in spatial proximity. Two aminoacids are considered close when their distance is between 4 and 8 Å. The binary adjacency matrix, having as rows and columns the residues ordered according to their position in the sequence, is the main information that is exploited when following this approach. Notice that in this unlabeled graph representation the different chemical properties of amminoacids are deliberately neglected. This representation highlights secondary order structures (i.e. α-helices and β-sheets) as patterns of the graph's adjacency matrix, as shown in Fig. 1.2. PCNs allow for a reliable reconstruction of the global protein structure [166] as well as

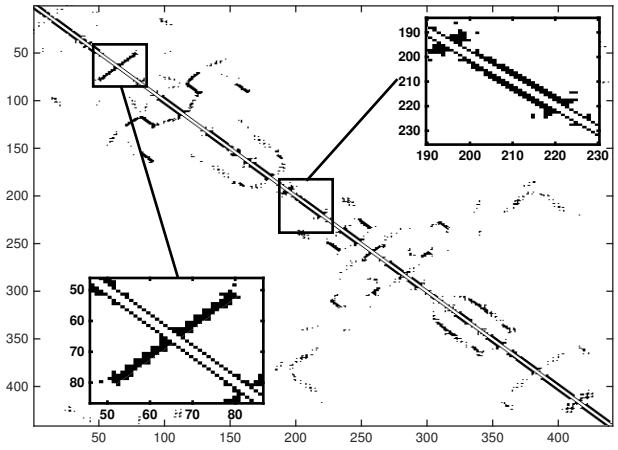


Figure 1.2. Secondary structure elements as motifs of a typical protein contact network represented by its adjacency matrix. Inset at the top right: α-helix; Inset at the bottom left: antiparallel β-sheet [169].

an efficient description of relevant biological properties of proteins such as allosteric effect and identification of active sites [51].

1.6.1 The Bartoli model

Generative models of networks are an important aspect of network theory. Indeed, a suitable network generative model that shows interesting/non-trivial properties has several advantages in network analysis, notably:

- provides insights on the wiring mechanisms leading to an observed emergent property
- serves as a null model for statistical analysis
- allows the sampling of new graphs
- allows the extrapolation of properties, i.e. sampling of graphs with properties that are unobserved in experimental data, like greater sizes, higher connectivities, etc.;

The design of a realistic generative model of protein contact networks is therefore an important topic for understanding the principles behind proteins structure and it will be thoroughly discussed in the following Chapters. The starting point we consider for generating a PCN is the model of Bartoli et al. [20]. In their work, the authors define a growth-based model for graph generation that consists of three simple rules. Starting from a graph of N vertices (i.e., amino acids):

1. A backbone is created by connecting adjacent neighbors up to a distance of 2, hence creating all edges $(i, i+1)$ and $(i, i+2)$. This corresponds to assigning contacts to the first two diagonals and, ensuring global connectivity of the network; in our works, however, we will consider a slight modification of this rule and we will allow only links between second-neighbors (i.e. vertices with distance 2 on the backbone)

2. Adding long range contacts between amino acids on the chain by selecting a pair (i, j) with a probability that decreases linearly with distance, i.e. $P_{\text{lin}}(i, j) \propto 1 - |i - j|$;

3. Connecting the residuals selected in the previous points and all their first neighbors. This corresponds to adding contacts for all combinations of pairs $\{i-1, i, i+1\} \times \{j-1, j, j+1\}$, where i and j are the previously selected vertices. This is motivated by the physical constraints of the chain, since putting in contact two aminoacids often causes other neighboring aminoacids on the sequence to come closer.

By iterating the points 2 and 3 of the previous rules, it is possible to generate a random synthetic PCN with a given connectivity.

Chapter 2

Heat diffusion on complex networks

In this chapter, we perform an extensive analysis of the mesoscopic organization principles of complex biological systems by analyzing different complex networks: protein contact networks, metabolic networks, and genetic networks, together with simulated networks created from generative models and utilized as reference. All considered networks are characterized in terms of two separate collections of numerical features. The first one is based on classical *topological descriptors*, such as the modularity and statistics of the shortest paths (see Sec. 1.2). The second one exploits the discrete heat kernel (HK), elaborated using the eigendecomposition of the normalized graph Laplacian (see Sec. 1.4). With a first preliminary analysis, we show that the different classes of networks are discriminated by a suitable embedding of these numeric features. This is reasonably expected, given the substantially different natures of the analyzed networks, but by no means can be considered as a trivial result. As a matter of fact, the distinction in terms of metabolic, genetic, and protein contact networks is based on network functions, and the demonstration of a link between functional and structural properties of the corresponding graph representation is a prerequisite for the soundness of the proposed strategy of analysis. An important result is that the two considered network characterizations resulted to be strongly correlated with each other, denoting an agreement in the two representations and providing a proof-of-concept of the reliability and interpretability of the adopted network descriptions. From this first analysis, it also emerged that protein contact networks display unique properties in terms of heat diffusion on network's topology, that do not allow for a straightforward classification in any of the considered models of networks, highlighting the need of further investigations on the peculiarities of these structures. The second and more important contribution elaborated in this chapter is the derivation of network-based heat diffusion properties that seem to agree, on principle, with known chemico-physical properties of protein molecules. Indeed, a computational analysis performed by exploiting the HK formalization demonstrates that a (simulated) diffusion process on protein contact networks proceeds slower than normal diffusion (i.e., we observe subdiffusion). Notably, a two-regime diffusion emerged from the analysis of the heat trace decay: a fast and a slow regime. The fast regime is driven by "shortcuts" putting in contact amino acid residues far-away along the sequence. Subdiffusion in proteins is a well-studied property describing energy flow [100–102, 149] and vibration dynamics [68, 130, 145, 146, 174], which has been investigated by several experiments. There is sufficient agreement on the fact that proteins, in their native structure, are highly modular and fractal networks [7, 17, 48, 52, 58, 104, 127]; yet they are characterized by suitable short paths connecting distant regions of the molecules responsible for the fast-track transport of energy and protein allosteric properties [100]. Here we observe also that, at odds with the other networks, the modularity of protein contact networks increases with the size of the network, a factor that also contributes to the subdiffusive property of their topology [64].

2.1 The Considered Networks

In this analysis we considered several sets of networks representing different real-world biological systems:

- 100 protein contact networks (PCN) extracted from randomly selected proteins of the E.Coli proteome. Such proteins have been obtained by integrating the Niwa *et al.* [135] E.Coli data with the available information of the respective native structures gathered from the Protein Data Bank repository [2]. The selected proteins are constituted by a number of amino acid residues ranging from 300 to 1000 units; for a detailed description of PCN see Sec. 1.6.

- 43 metabolic networks (MN) describing organisms belonging to all three domains of life. Vertices of such networks are the substrates and product of the chemical reactions, while the edges are the reactions catalyzed by the enzymes. As shown in Ref. [90], these large metabolic networks exhibit a typical scale-free topology. The sizes of the networks ranges from 300 to 1500 vertices.

- 50 realistic gene regulatory networks (GEN) with a number of vertices varying from 200 to 1100 genes/vertices. The GEN networks are generated with the SysGenSIM software [143], a MATLAB™toolbox for the simulation of systems genetics datasets. Artificial networks and data by SysGenSIM have already been officially employed for the verification of gene network inference algorithms, such as in the DREAM5 Systems Genetics challenge [1]; they have also been used as benchmarks for the development of state-of-the-art reverse-engineering algorithms [62, 142]. GEN networks have been generated with the Exponential Input Power-law Output (EIPO) model, i.e., they are built by (i) sampling the number of ingoing and outgoing edges for each vertex from, respectively, an exponential and a power law distribution, and then by (ii) connecting the vertices accordingly. These artificial EIPO networks exhibit two well-known structural characteristics of real gene networks: modularity [19], and the vertex in-degree and out-degree distributions fitting, respectively, an exponential and a power law curve [71]. Besides the adopted EIPO topology, we considered an average vertex degree varying from 4 to 8: the average degree has been sampled in such a wide range due to the uncertainty in the size of the interactome in typical gene regulatory networks. Apparently, the complexity of biological organisms better correlates with the number of interactions between genes than with the number of genes. Therefore, the average number of edges in gene regulatory networks varies according to the complexity of the represented organism [157]; it makes then sense to study gene regulatory networks with a different number of interactions/edges.

To obtain suitable references with the aim of helping us in discussing the results, we considered 130 additional networks of varying size belonging to well-known classes of graphs. Such networks play the role of "probes" in the space of topological descriptors. In particular, we considered 10 Erdős-Rényi (ER) graphs generated with probability $p = \log(n)/n$; 10 Barabási-Albert (BA) scale-free networks [18] with a six-edges preferential attachment scheme; and 10 random regular graphs (REG) with degree equal to six. To cover all network sizes, such probe networks are generated with a number of vertices ranging from 200 to 1100. Finally, we also generated the synthetic counterpart of the 100 real proteins (denoted as PCN-S in the following, see Sec. 1.6.1). Such synthetic proteins have been generated by considering the same number of vertices and edges of the real proteins. The generation mechanism of the topology follows the three-rule scheme proposed in Ref. [20], to simulate the folded configuration of the protein backbone by a probability of contact decreasing with the sequence distance. The only exception

is for the rule involving edges of the backbone structure. In fact, to be consistent with the architecture of our real proteins (we considered edges among residues within 4–8 Å), in PCN-S we added edges only among consecutive residues in the sequence having distance 2. It is worth pointing out that such a generation mechanism gives rise to networks with typical small world topology [20].

2.2 Characterization of the Graph Topology

In the following, we consider two different characterizations of the considered networks. In the first characterization we describe each network with a vector whose components are the values of several topological descriptors (TD), directly elaborated from its topology. We consider the number of vertices (V) and edges (E) as basic descriptors of the size and connectivity of the network; the modularity (MOD, as defined in Sec. 1.2.2) for quantifying the presence of a global community/cluster structure. The numerical value of the modularity is evaluated on the best partition of nodes obtained with the Louvain algorithm [27]; the average closeness centrality (ACC, Sec. 1.2.1), average shortest path (ASP, Sec. 1.2.4), average degree centrality (ADC, Sec. 1.2.1), and average clustering coefficient (ACL, Sec. 1.2.3) [40]; the energy (EN) and Laplacian energy (LEN) of the spectrum (as defined in Ref. [73]), that are related to the spectral properties of the adjacency and laplacian matrix; two invariant features from the heat kernel, respectively the heat trace at time $t = 5$ (corresponding to a transient regime in the diffusion of heat) and the first coefficient of the heat content invariants ($m = 1$); the graph ambiguity (A, as defined in [108]), which expresses the degree of irregularity of the topology; finally, the entropy of a stationary Markovian random walk on the graph topology (H) [47].

The second characterization is composed by three sets of features extracted from the heat kernel properties (see Sec. 1.4) of the networks: the heat trace (HT), the heat content (HC), and the heat content invariants (HCI). Notice that HT and HC are time-dependent characteristics, while HCI is not. Therefore, in this characterization we consider the series HT and HC for the instants $t = 0, 1, 2, ..., 9$ and the series HCI for $m = 0, 1, 2, ..., 9$. Given these two characterizations, we proceed to calculate the 4 embedding vectors for each network, i.e. the topological descriptors vector, the heat trace vector, the heat content vector, and the heat content invariants vector. For each network, each of these vectors is to be considered in a separate space.

2.3 Results

In the following we present the results of the embedding of each network in each of the embedding spaces, that is, TD, HT, HC, and HCI. In order to simplify the analysis and highlight relevant patterns, for each of the 4 spaces we perform a dimensionality reduction by means of a *Principal Component Analysis* [91], which allows to choose the directions of maximum variance for a given dataset of points. In this way, the points representing the networks are projected in the space spanned by the principal components (PC), that are by construction orthogonal to each other.

2.3.1 Analysis of Topological Descriptors

Fig. 2.1 shows three different projections and a 3D visualization of the PCA of the topological descriptors (PCA-TD). The first three PCs are sufficient to explain more than 90% of the variance ($\simeq 91\%$). As it is possible to observe in Fig. 2.1(a), PC1-PC2 offer a very clear discrimination among the different classes of networks. The separability persists also when considering

PC1-PC3, while however we observe that GEN lose compactness and overlap with MN. By considering PC2-PC3, instead, PCN overlap with REG. However, the overall picture emerging from PCA-TD clearly points to the possibility of distinguishing the network types .

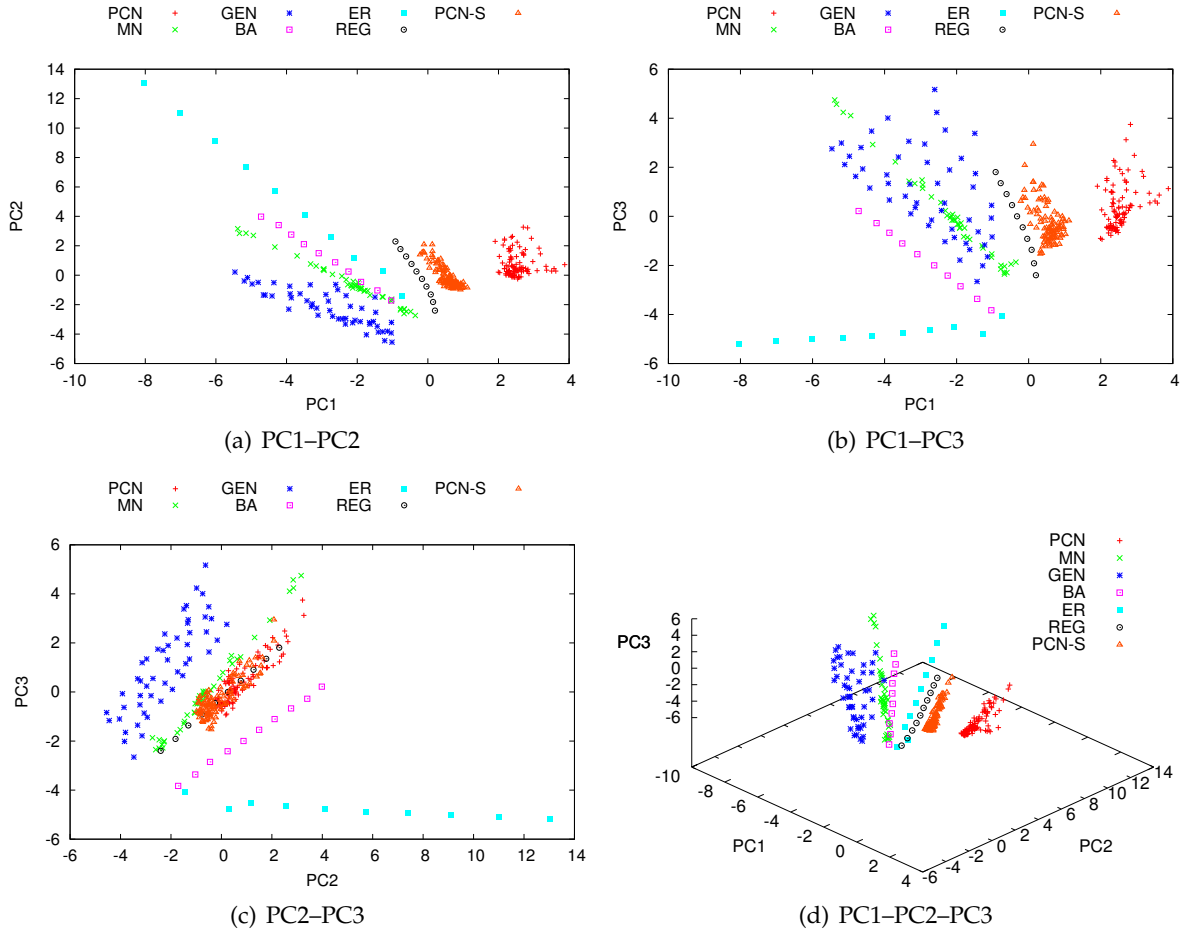
Let us now interpret such PCs. Tab. 2.1 shows the loadings of the first three factors of PCA-TD. The first factor (FACTOR-1) is primarily characterized by MOD, ACC, and ASP. As MOD increases (the community structure becomes more evident) the length of preferential paths connecting different regions of the network increases as well. Indeed, ACC and ASP are, respectively, negatively and positively correlated with MOD. It is worth mentioning that ACC and ASP offer a somewhat opposite view of the same feature, i.e., the efficiency of the paths in the networks. As the global community structure emerges (captured by MOD), also the local clustering structure (ACL) increases as well, although ACL is less loaded on FACTOR-1. In addition it is worth noting the agreement among the random walk entropy (H) and the modularity: predictability of stationary random walks is affected by the presence of network modules/communities. FACTOR-2 positively correlates the number of vertices V with LEN, which clearly points to the correlation among the network size in terms of number of vertices and the global architecture. The meaning of this factor will appear more clear in Sec. 2.3.4, where we will discuss the scaling of the number of vertices with MOD and the invariant characteristics of the HK. Finally, the third factor (FACTOR-3) could be interpreted as the "redundancy" of the network wiring substrate. In fact, descriptors heavily loaded on FACTOR-3 are those more directly related to the adjacency matrix–edges. The ambiguity (A) decreases as the number of edges increases. This means that adding redundancy to the network (i.e., alternative paths) affects the regularity of the topology. It is immediate to recognize how the different types of networks are characterized by local linear models in the globally orthogonal PC spaces. These linear models correspond to different scaling relations with network size – discussed later in Sec. 2.3.4. A simple look at Fig. 2.1 allows to catch the singular position of PCN on the extreme right of the most informative PC1-PC2 space, therefore hinting at the peculiar character of PCN with respect to classical network architectures. Moreover it is worth noting that the synthetic networks PCN-S are the most similar to PCN, although it is not possible to appreciate any overlap. This fact suggests that proteins are not just "coiled strings" as hypothesized in Ref. [20]. In addition to the features coming from the folding of a continuous backbone, PCN have other peculiar characteristics.

Table 2.1. Loadings of the first three factors of PCA-TD. Relevant correlations are in bold.

DESCRIPTOR	FACTOR-1	FACTOR-2	FACTOR-3
V	-0.0441	**0.9953**	0.0513
E	-0.2053	0.5095	**0.8082**
MOD	**0.9591**	-0.1383	-0.1036
ADC	-0.2294	0.0428	**0.9375**
ACC	**-0.9918**	0.0353	0.0637
ASP	**0.9281**	-0.0279	0.0315
ACL	0.6716	-0.3588	-0.0010
EN	-0.0166	0.6830	**0.7268**
LEN	-0.3944	**0.8407**	0.1712
HT ($t = 5$)	0.6696	0.6486	-0.1007
HCI ($m = 1$)	0.4914	-0.6172	0.4639
H	0.6906	-0.2774	0.4918
A	-0.3229	0.1878	**-0.7584**

2.3.2 Analysis of the Heat Kernel

We consider three types of invariant features elaborated from the HK: HT, HC, and HCI. As mentioned in Sec. 2.2, for the PCA of HT and HC we take into account 10 time instants going

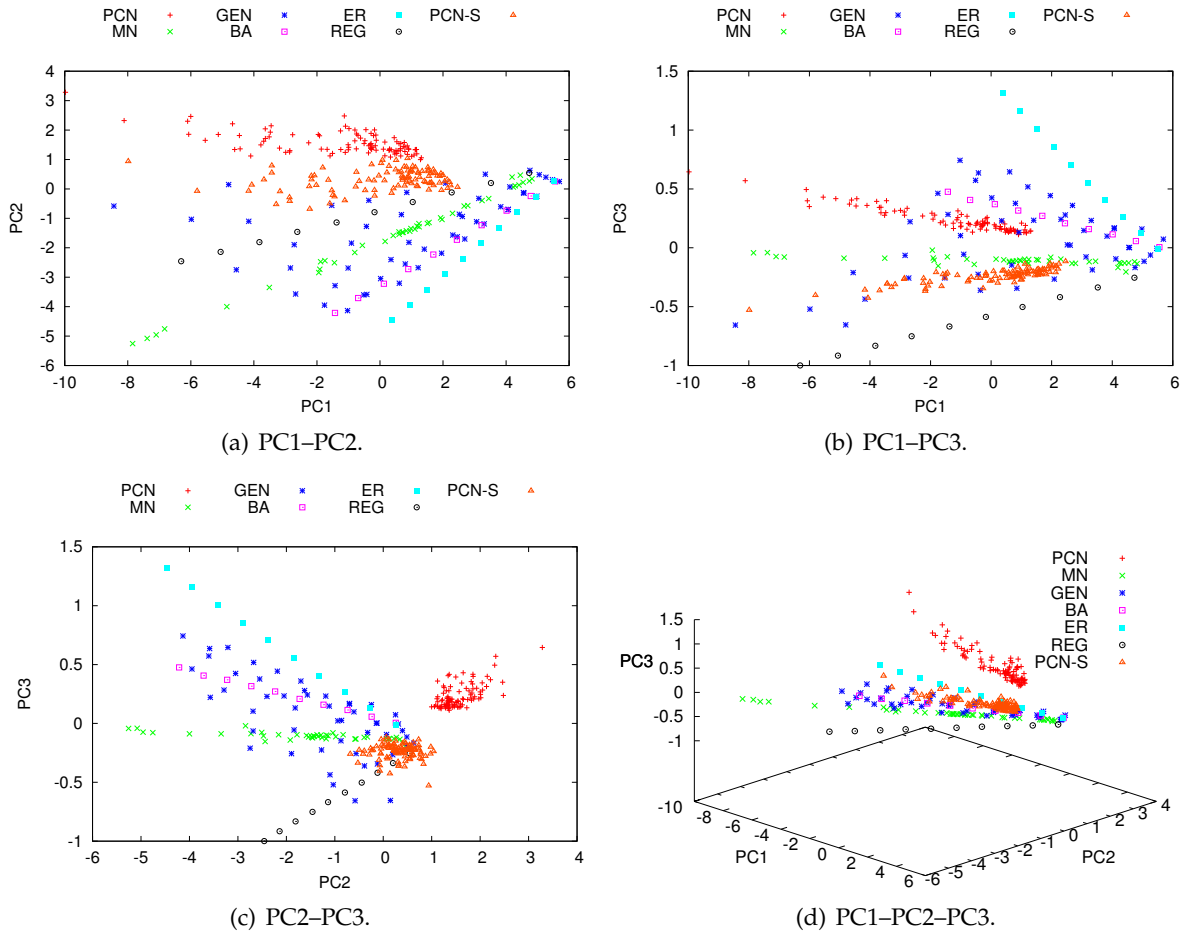


(a) PC1–PC2 (b) PC1–PC3

(c) PC2–PC3 (d) PC1–PC2–PC3

Figure 2.1. Embedding considering the first three PCs of PCA-TD.

from $t = 0$ to $t = 9$, while for the PCA of HCI we consider the first ten coefficients q_m of the series of Eq. (1.31). In all cases, the first three PCs are sufficient to explain more than 90% of the variance of the original data, and so they are retained for the embedding. In Fig. 2.2 it is shown the PCA of the HT representation (PCA-HT). From PC1-PC2 and PC2-PC3 of PCA-HT it is possible to observe that PCN are quite clustered and well-separated from the other networks when considering their HT, while GEN depict an incoherent pattern (this is valid for all three PCs). In Fig. 2.3, instead, we show the PCA of the HC representation (PCA-HC). We remind to the reader that HT and HC are correlated, since HC considers the information provided by both eigenvalues and eigenvectors of the normalized Laplacian, and not just the eigenvalues as in the HT case. From PCA-HC it is possible to note that all sets of networks denote a clear distinguishability; considering either PC1-PC2 and PC2-PC3 almost all networks seems to denote a very peculiar configuration in the PCA space. Finally, in Fig. 2.4 we show the PCA of the HCI representation (PCA-HCI). From the PCs of PCA-HCI we observe that PCN, REG, and ER denote a very compact configuration in the PCA-HCI space, while GEN, BA, and MN present a more sparse distribution. This fact might be interpreted by observing that such two groups differentiate among networks having a clear scale-free topology (second group) and those that are not scale-free (first group). Interestingly, PCN-S seem to lie in-between those two groups.



(a) PC1–PC2. (b) PC1–PC3.

(c) PC2–PC3. (d) PC1–PC2–PC3.

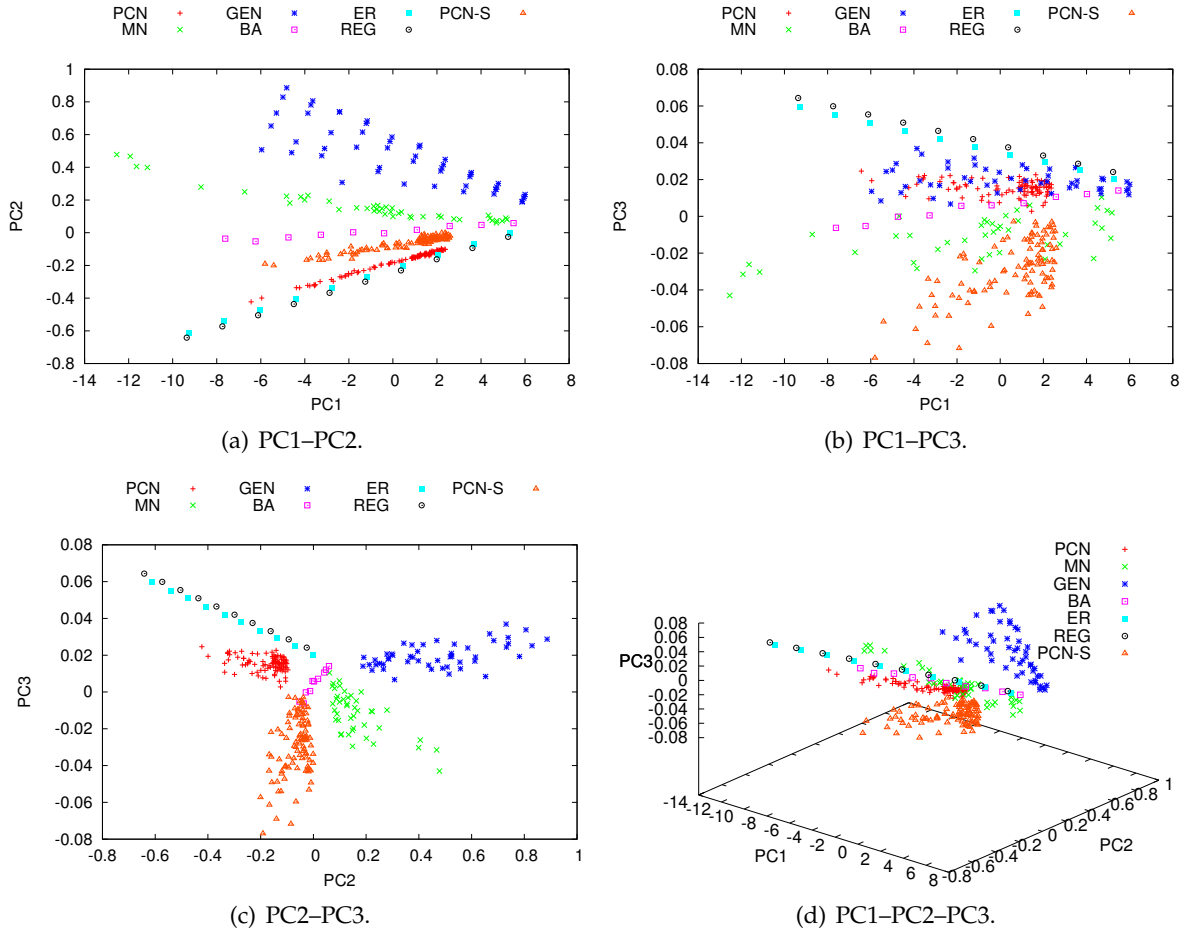
Figure 2.2. Embedding of the first three PCs of PCA-HT considering $t = 0, 1, ..., 9$.

2.3.3 Canonical Correlation Analysis of the PCA Representations

In this Section we discuss the canonical correlation analysis (CCA) calculated among the various PCA spaces presented in the previous sections. The canonical correlation analysis provides a measure of agreement in description between two spaces, so we exploit this methodology to assess to what extent the HK spaces described above are a meaningful representation of the networks' topology. For the CCA, we always consider the first three PCs of each representation. In Tab. 2.2 are reported the pairwise correlation values among the most important canonical variates. There is a strong agreement among all the considered PCA representations. Since part of the information from the HK is present also in TD and may lead to spurious correlations, we considered also a PCA representation of TD that does not include such information, in the table it is indicated as "PCA-TD_NO-HK". Interestingly, removing the information of the HK from the TD does not alter the scored correlation, so giving a demonstration of the strong coherence between TD and HK based representations of the considered networks. This result suggests the possibility to interpret the three HK based representations as alternative descriptions of the networks space that are in agreement with the TD representation.

2.3.4 Scaling and Heat Diffusion Analysis

We now proceed to study the networks' properties in terms of scaling of MOD, HT, HC, and HCI with respect to the network size. Fig. 2.5 shows the dependence of the modularity with



(a) PC1–PC2. (b) PC1–PC3.

(c) PC2–PC3. (d) PC1–PC2–PC3.

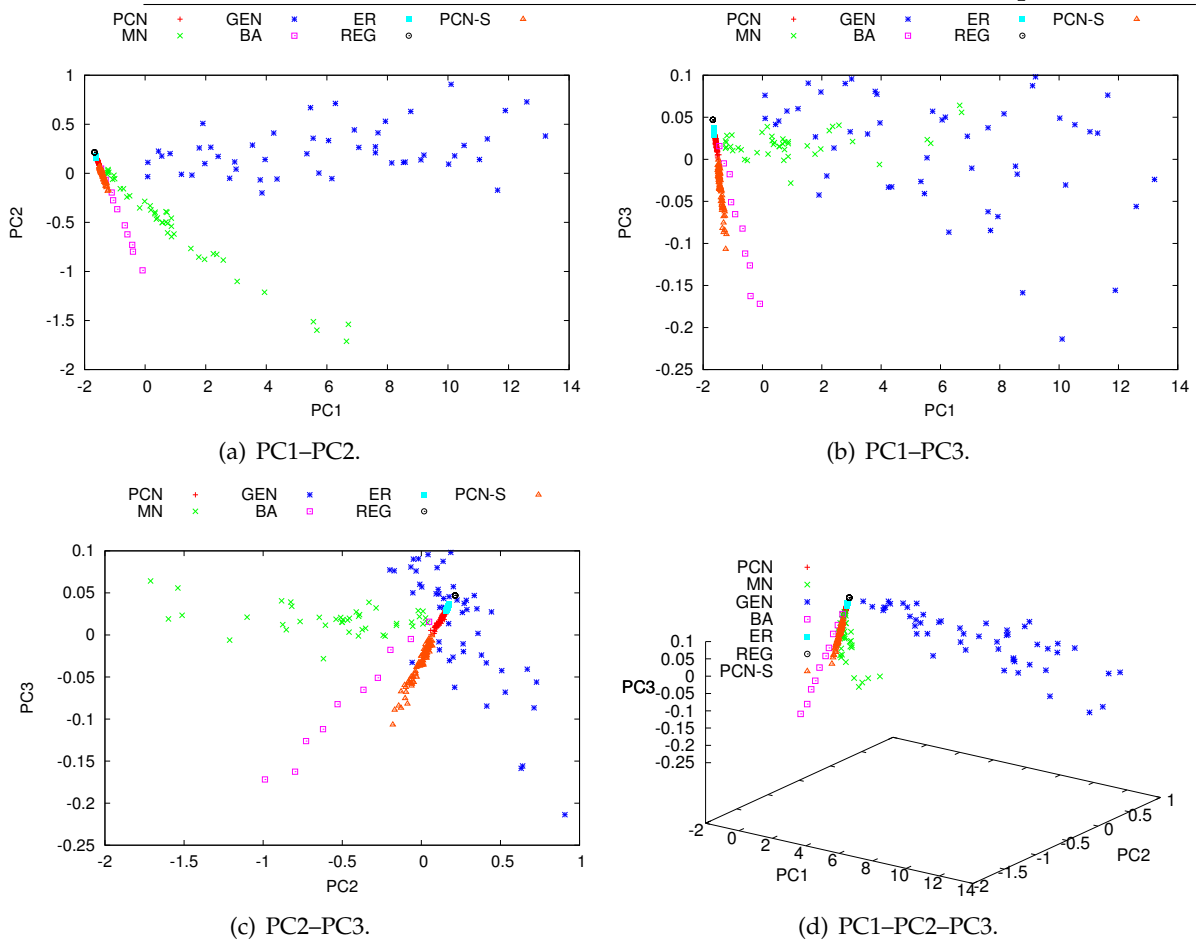
Figure 2.3. Embedding of the first three PCs of PCA-HC considering $t = 0, 1, ..., 9$.

Table 2.2. Canonical correlation coefficients between the first canonical variates relative to different principal component spaces.

	PCA-HT	**PCA-HC**	**PCA-HCI**
PCA-TD	0.993	0.992	0.961
PCA-TD_NO-HK	0.988	0.986	0.946

the size of the networks, where the linear fitting are introduced to highlight the increasing or decreasing trend of the depence. As already noted in Tab. 2.1, V and MOD do not appear to be globally correlated. In fact, PCN and PCN-S are the only architectures that show an increasing trend, while the others appear to be almost independent. We note an exception for ER that tend toward a negative correlation, which agrees with the analytical results on the modularity of ER [72]. It is worth observing more in detail the particular dependence pattern of GEN, which does not show a clear trend. In Fig. 2.5(b) we show the scaling for GEN by considering the different average degrees used for the EIPO model, where we can observe that each average degree gives rise to a definite trend of MOD.

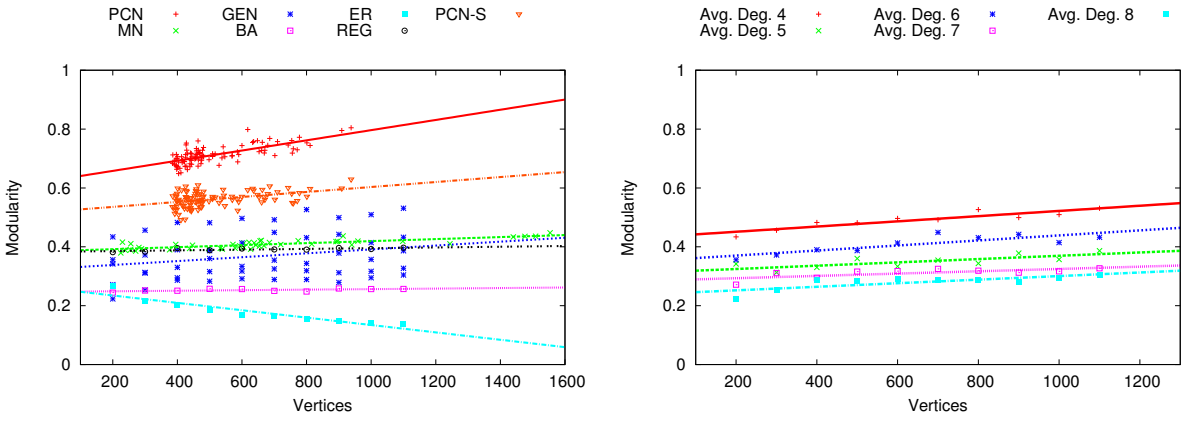
Figs. 2.6, 2.7, and 2.8 show the scaling of all considered HK invariants. Initially we consider only three relevant time instants for HT, i.e., $t = 1, 5, 9$, which are depicted, respectively, in Figs. 2.6(a), 2.6(b), and 2.6(c). It is possible to observe that, as expected, at $t = 1$ all networks show a similar increasing linear trend with respect to the network size. As the time increases,



(a) PC1–PC2. (b) PC1–PC3.

(c) PC2–PC3. (d) PC1–PC2–PC3.

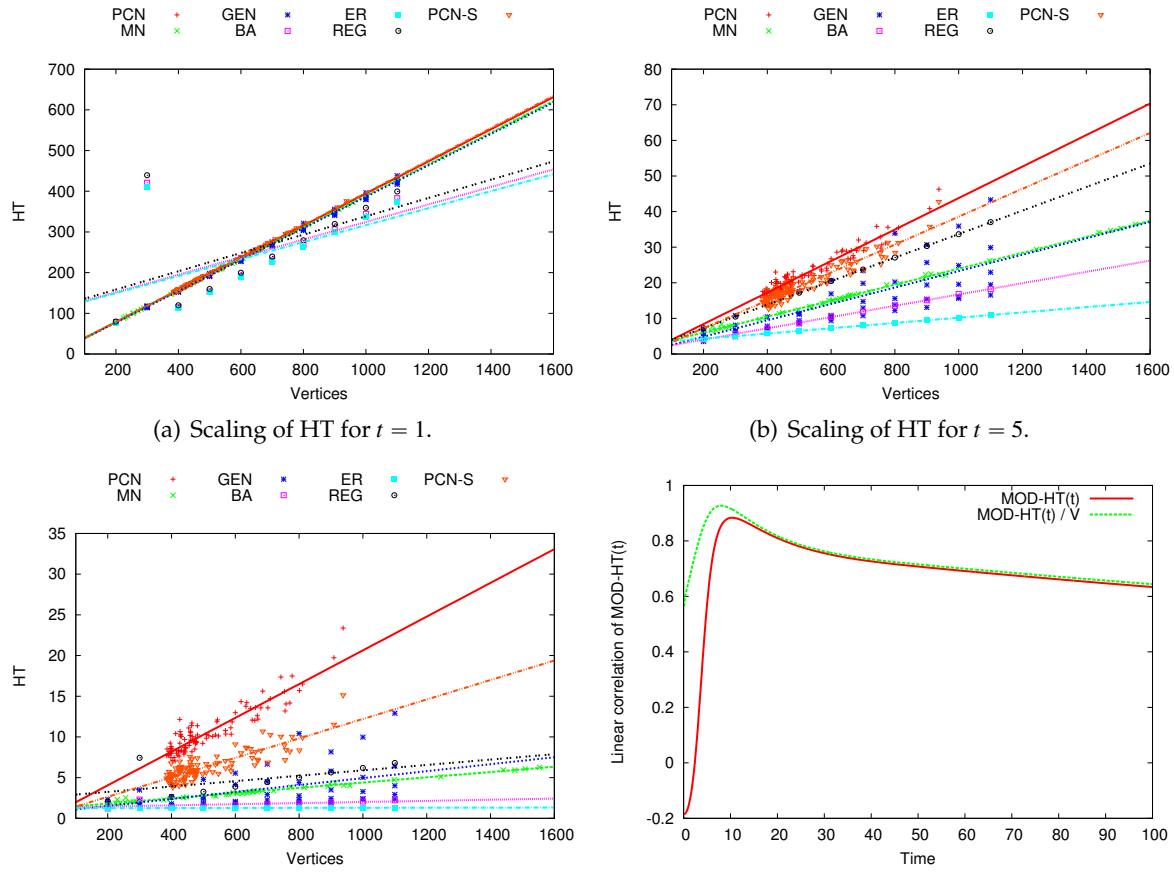
Figure 2.4. Embedding of the first three PCs of PCA-HCI considering the first ten HCI coefficients.



(a) Scaling of modularity. (b) Modularity of GEN with different average degree.

Figure 2.5. Dependence of modularity over network size. The linear fittings highlight the trends of the dependences for each set of network.

instead, PCN show a positive slope at least one order of magnitude greater than the others. At first, this fact might be attributed solely to the intrinsic high modularity characterizing the protein structures. To this end, in Fig. 2.6(d) we globally correlated MOD with HT over



(a) Scaling of HT for $t = 1$.

(b) Scaling of HT for $t = 5$.

(c) Scaling of HT for $t = 9$.

(d) Linear correlations among MOD and HT(t).

Figure 2.6. Scaling of HT over networks size.

time – the time t here has a fine-grained sampling going from 0.1 to 100 with an increment step of 0.1. In the same plot, we show also the partial correlation obtained when considering the number of vertices as the control variable (indicated as "MOD–HT(t) / V" in the figure). The linear correlation trend shows that initially the two quantities are fairly anti-correlated, while they soon become very correlated, reaching the maximum correlation ($\simeq 0.88$) around the time instant $t = 10$. Successively, the correlation decreases with a smooth trend. The partial correlation demonstrates that the initial negative correlation is due to the effect of the network size; correlations are positive when the size is removed. This variability in the correlation points out the fact that the nature of information provided by HT is consistent with the one provided by MOD, although they are by no means equivalent. The diffusion of heat on the graph is indeed highly dependent of the modular structure of the network. Intuitively, a modular structure slows down the diffusion process and this is reflected in the heat trace. Notably, as we show in the following, the heat trace offers a richer type of information.

2.4 Ensemble Heat Trace

As explained in Sec. 1.4, the heat trace of a graph is a function that describes the diffusion of heat on the network topology and depends on the distribution of the normalized laplacian eigenvalues $\tilde{\lambda}_i$. However, such a characterization is cumbersome to handle for a set of n graphs, since it involves n different functions of time. In order to design a more compact characterization

that describes the heat diffusion of a whole set of networks, we consider the linear best-fitting obtained from the HT scaling functions considered in the previous Section. Indeed, as it is possible to observe in Figs. 2.6(a), 2.6(b), and 2.6(c), at a fixed time t and for a given set of graphs, the HT as a function of the graph size follows an approximately linear trend. In order to justify this observation, consider the heat trace of a generic graph G of size N at a fixed time t:

$$\mathrm{HT}_G(t;N) = 1 + \sum_{i=2}^{N} \exp(-\lambda_i t). \tag{2.1}$$

where λ_i are the eigenvalues of the normalized Laplacian of G. Let us define an *ensemble* of graphs $\mathscr{C}$, i.e. a set of graphs that share a common characteristic spectral density. Such spectra can be synthetically described by considering the spectral density of the ensemble $\mathscr{C}$. Accordingly, we can consider the eigenvalues as i.i.d. random variables, $\tilde{\lambda}_i$, assuming values according to the spectral density of the ensemble, except for $\tilde{\lambda}_1$ that assumes deterministically the value 0. The HT of a generic graph $G \in \mathscr{C}$ of dimension N can be written as

$$\mathrm{HT}_G(t;N) = 1 + \sum_{i=2}^{n} \exp(-\tilde{\lambda}_i t) = 1 + \sum_{i=2}^{N} \exp(-\tilde{\lambda} t). \tag{2.2}$$

where last step is carried out by considering that, since the $\tilde{\lambda}_i$ are assumed i.i.d., their realizations can be expressed as N realizations of a single random variable $\tilde{\lambda}$. For fixed time t, we can define the *ensemble heat trace*, $\mathrm{HT}_{\mathscr{C}}(N;t)$, as the mean heat trace over all graphs of the ensemble $\mathscr{C}$ with size N at fixed time t

$$\mathrm{HT}_{\mathscr{C}}(N;t) = \langle \mathrm{HT}_G(t;N) \rangle_{\mathscr{C}} = 1 + \sum_{i=2}^{N} \langle \exp(-\tilde{\lambda} t) \rangle_{\mathscr{C}} = 1 + (N-1) \langle \exp(-\tilde{\lambda} t) \rangle_{\mathscr{C}}, \tag{2.3}$$

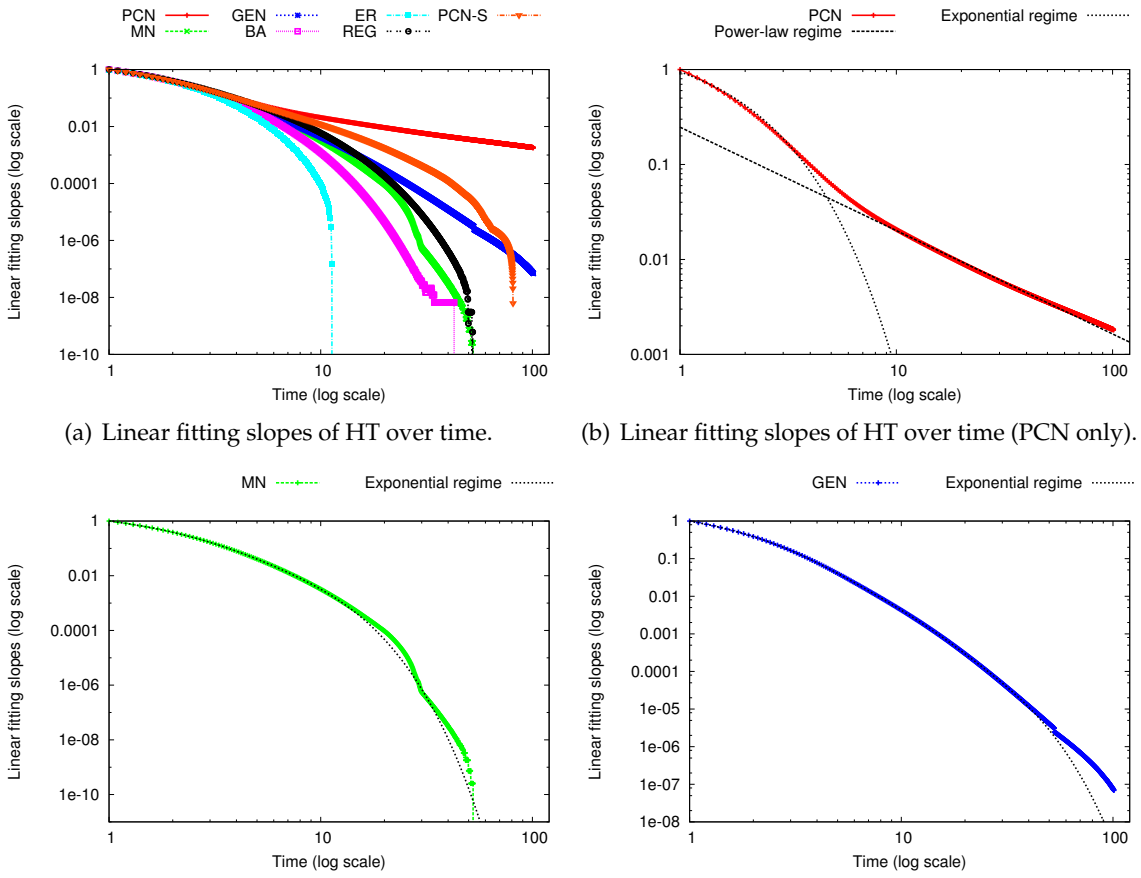
Since the term $\langle \exp(-\tilde{\lambda} t) \rangle_{\mathscr{C}}$ does not depend on the size of the graph N, $\mathrm{HT}_{\mathscr{C}}(N;t)$ can be expressed as a linear function of the graph size

$$\mathrm{HT}_{\mathscr{C}}(n;t) = 1 - \alpha_{\mathscr{C}}(t) + \alpha_{\mathscr{C}}(t) \cdot n \simeq \alpha_{\mathscr{C}}(t) \cdot n, \tag{2.4}$$

where $\alpha_{\mathscr{C}}(t) = \langle \exp(-\tilde{\lambda} t) \rangle_{\mathscr{C}} \in [0,1]$ is a time-dependent angular coefficient (slope) that is characteristic for the entire ensemble $\mathscr{C}$. By fitting linearly the HT we therefore implicitly hypothesize the possibility to consistently describe each class of networks with an ensemble, $\mathscr{C}$, characterized by a unique probability density function of the (normalized) Laplacian eigenvalues (see Ref. [123] for a related theoretical study). This assumption is also justified by the results of PCA-HT reported in Fig. 2.2, which show good agreement among the networks of the same class. As a consequence, the linear best-fitting of the HT as a function of the graph size allows us to consider a statistic over an entire homogeneous class of networks, instead of focusing on each isolated network dynamics separately. It is straightforward to realize that $\mathrm{HT}_{\mathscr{C}}(N;t) = N$ for $t = 0$, i.e., $\alpha_{\mathscr{C}}(0) = 1$. As t grows $\alpha_{\mathscr{C}}(t)$ decreases with a rate that is related to the characteristic HT decay of the ensemble.

In Fig. 2.7(a) we show the linear best-fitting slopes of HT, $\alpha_{\mathscr{C}}(t)$, as a function of time – note that t always varies from 0 to 100 with an increment step of 0.1. While one expects to observe trends consistent with an exponential decay (see definition of HT in Eq. 1.28), it is possible to recognize a different trend for the PCN ensemble. For the sake of a better visualization, in Figs. 2.7(b), 2.7(c), and 2.7(d) we report the same plot but isolating, respectively, PCN, MN, and GEN; other networks are omitted for the sake of brevity. Fig. 2.7(b) depicts what we might consider a change of functional form for the PCN trend at some point in time (i.e., starting

around $t \simeq 5$). This change of regime in the diffusion lasts few time instants, then the trend switches from exponential to power law like. This is not observed on the other networks that, instead, remain consistent with an exponential decay. In practice, for $\widetilde{t} > 5$, the diffusion in PCN seems to be consistent with a power law, $\alpha_{\mathscr{E}}(\widetilde{t}) \sim \widetilde{t}^{-\beta}$, where in our case the characteristic exponent is $\beta \simeq 1.1$. Similar anomalies of functional form have been observed in the (cumulative) distribution of many experimental time series, especially in those related to financial markets [98]. This phenomenon might happen when the functional form is consistent with one of the q-exponentials family, which originated in the field of non-extensive statistical mechanics [161]. In the case of PCN, this behavior is the signature of a crucial physical property of proteins, i.e., the efficient yet controlled energy flow between different areas of the structure. Energy flow in proteins mimics the transport in a three-dimensional percolation cluster [100]: energy flows readily between connected sites of the cluster and only slowly between non connected sites. This experimentally validated double regime seems to be captured by the HT decay trend shown in Fig. 2.7(b). This result is elaborated from a minimalistic PCN model, so confirming the relevance of this graph-based representation in protein science [50].



(a) Linear fitting slopes of HT over time.

(b) Linear fitting slopes of HT over time (PCN only).

(c) Linear fitting slopes of HT over time (MN only).

(d) Linear fitting slopes of HT over time (GEN only).

Figure 2.7. Scaling of HT linear best fitting slopes over time (time is sampled in 1000 equally-spaced points between 0 and 100).

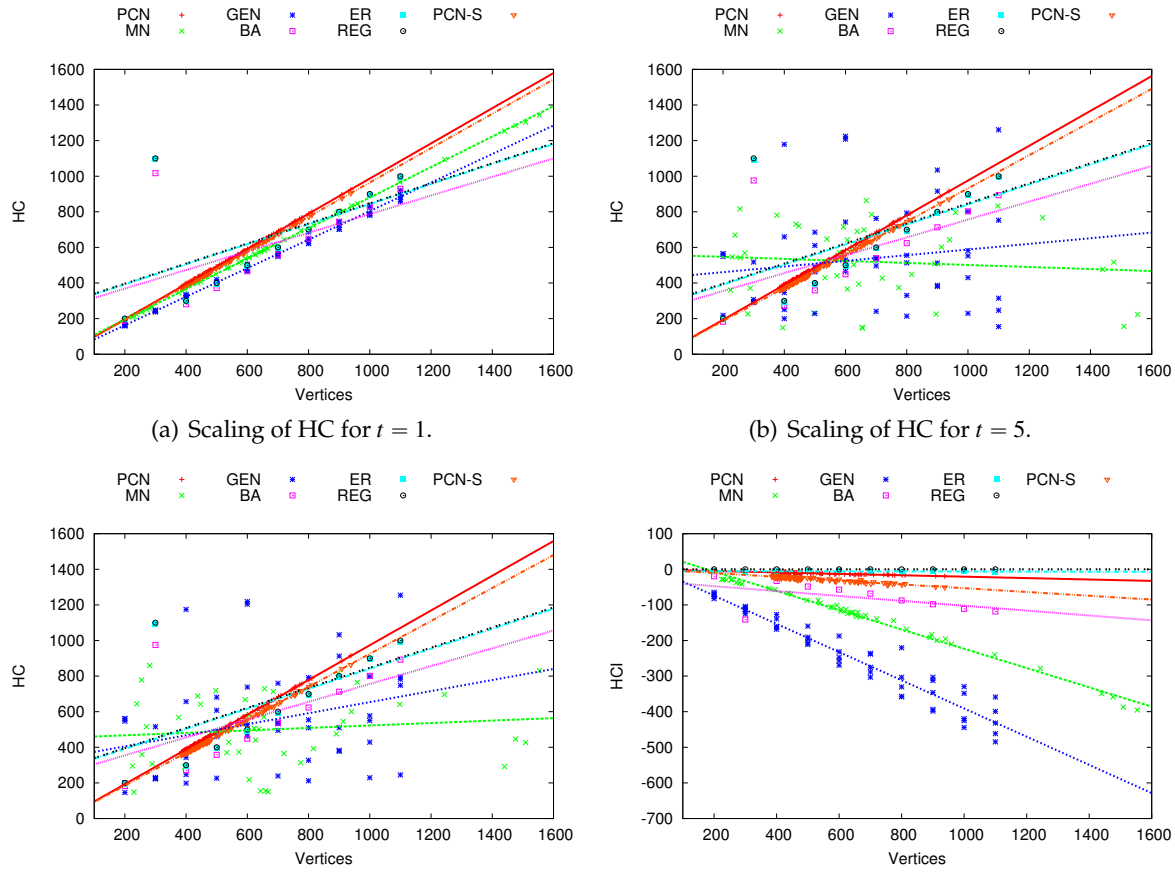
Now let us consider the results for the HC (Figs. 2.8(a), 2.8(b), and 2.8(c)). Those three figures depict the scaling of the HC over the network size, considering the information of the entire HM. Notably, PCN and PCN-S are the only network types showing a consistent linear scaling with the size for all time instants. Other networks are not well-described by a linear fit as

the time increases. Finally, in Fig. 2.8(d) we show the scaling of the first HCI coefficient with the vertices (please note that for $m = 1$, Eq. 1.34 yields negative values). Of notable interest is the fact that PCN denote a nearly constant trend. This means that, since the HCIs are time-independent features synthetically describing the HC information, PCN denote a similar characteristic in this respect, as in fact HC scaling in Fig. 2.8 is consistently preserved over time.

In Fig. 2.9 we offer a visual representation of the heat diffusion pattern over time that is observable through the entire HM. We considered two exemplar networks of exactly the same size: the "JW0058" protein and the synthetic counterpart belonging to PCN-S that we denote here as "JW0058-SYNTH". As discussed before, PCN are characterized by a highly modular and fractal structure, while the considered synthetic counterpart exhibits a typical small world topology. Accordingly, by comparing the diffusion occurring on the two networks over time, it is possible to recognize significantly different patterns that were not noted in the scalings of Fig. 2.8. Of course, initially ($t = 1$) the heat is mostly concentrated in the vertices, which results in a very intense trace. As the time increases, the diffusion pattern for the real protein is more evident and also persistent. This is in agreement with recent laboratory experiments [100, 101], which demonstrated that diffusion in proteins proceeds slower than normal diffusion. In graph-theoretical terms, this means that the spectral gap considerably dominates the sum in Eq. (1.28) as t becomes large. On the other hand, the diffusion for JW0058-SYNTH is in general faster since in fact the trace vanishes quickly. It is worth noting the difference in intensity that emerges from the figures. This fact is due to the different architectures characterizing the two networks: PCN are considerably more modular than PCN-S. We obtained analogue results by considering the other network types.

2.5 Discussion

In this chapter we have investigated the structure of three types of complex networks: protein contact networks, metabolic networks, and gene regulatory networks, together with simulated archetypal models acting as probes. We biased the study on protein contact networks, highlighting their peculiar structure with respect to the other networks. Our analysis focused on ensemble statistics, that is, we analyzed the features elaborated by considering several instances of such networks. We considered two main network characterizations: the first one based on classical topological descriptors, while the second one exploited several invariants extracted from the discrete heat kernel. We found strong statistical agreement among those two representations, which allowed for a consistent interpretation of the results in terms of principal component analysis. Our major result presented in this paper was the demonstration of a double regime characterizing a (simulated) diffusion process in the considered protein contact networks. As shown by laboratory experiments, energy flow and vibration dynamics in proteins exhibit subdiffusive properties, i.e., slower-than-normal diffusion [100]. The notable difference in the diffusion pattern between real proteins and the herein considered simulated polymers (whose contact networks have the same local structure of the corresponding real proteins), points to a peculiar mesoscopic organization of proteins going beyond the pure backbone folding. The observed correlations between MOD and HT indicates this principle in the presence of well-characterized domains. The novelty of our results is that we were able to demonstrate such a well-known property of proteins by exploiting graph-based modeling and computational tools only. The fact that the observed properties emerged with no explicit reference to chemico-physical characterization of proteins, relying hence on pure topological properties only, suggests the existence of general universal mesoscopic principles fulfilling the hopes expressed by Laughlin *et al.* [99].



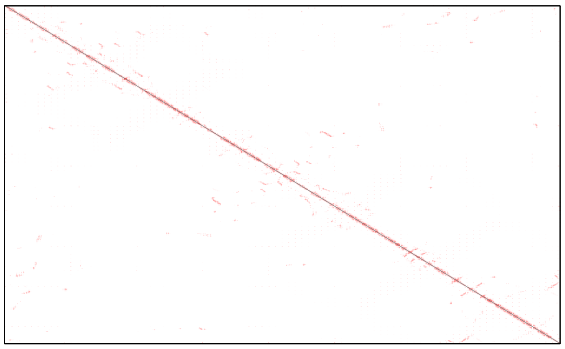
(a) Scaling of HC for $t = 1$.

(b) Scaling of HC for $t = 5$.

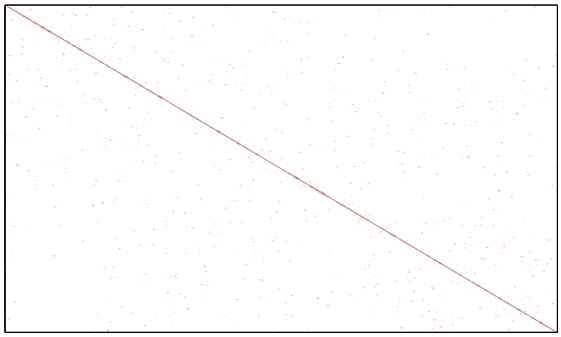
(c) Scaling of HC for $t = 9$.

(d) Scaling of HCI (first coefficient).

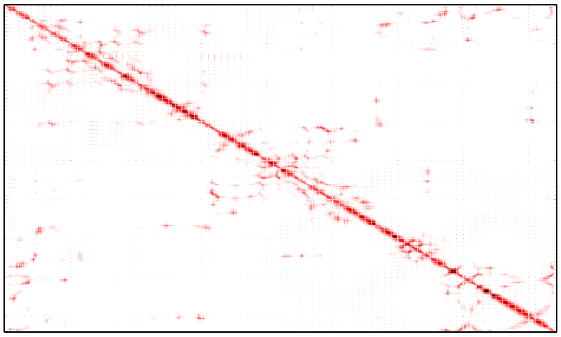
Figure 2.8. Scaling of HC and HCI over network size.



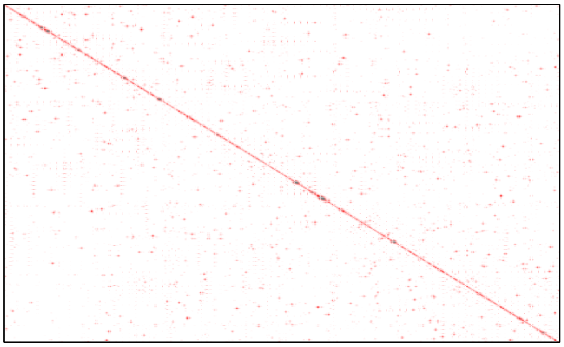
(a) HM for JW0058 at $t = 1$.



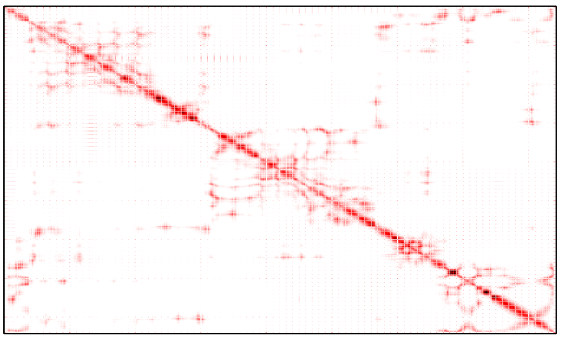
(b) HM for JW0058-SYNTH at $t = 1$.



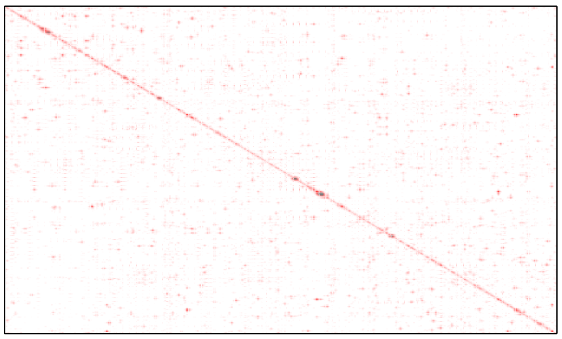
(c) HM for JW0058 at $t = 5$.



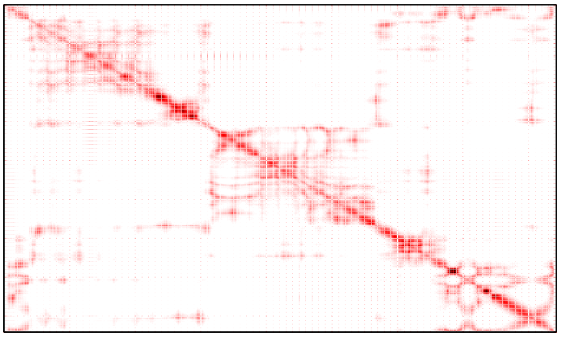
(d) HM for JW0058-SYNTH at $t = 5$.



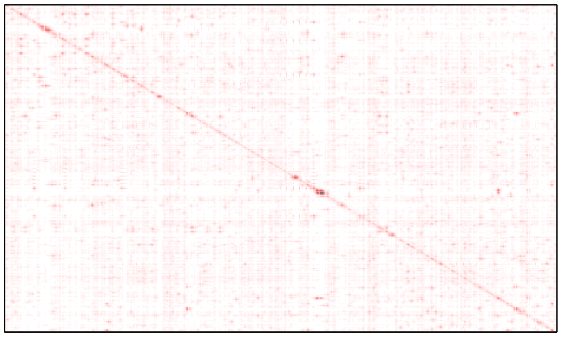
(e) HM for JW0058 at $t = 9$.



(f) HM for JW0058-SYNTH at $t = 9$.



(g) HM for JW0058 at $t = 15$.



(h) HM for JW0058-SYNTH at $t = 15$.

Figure 2.9. HM diffusion pattern over time for the real JW0058 protein and its synthetic counterpart.

Chapter 3

Network as a time series

In this chapter we exploit the Multifractal Detrended Fluctuation Analysis (MFDFA) [23, 94, 136], a generalization of the Detrended Fluctuation Analysis (DFA) [139], to study time series obtained from complex networks via stationary unbiased random walks (RW). The time series consist of the successive measurements of a given observable of the current node the walker is visiting, as described in Sec. 1.5. Our aim is to discover the existence of long-range correlations in these generated time series by means of the MFDFA. The MFDFA builds upon a generalization of the so-called Hurst exponent as a detector of long-range correlations [21, 153]. At the basis of Hurst exponent is the idea of characterizing time series in terms of their degree of *persistence*: roughly speaking, a series is long-range correlated (persistent) if the underlying process has memory of the past states, a property that is firstly noticeable as a heavy-tail in the corresponding autocorrelation function. Brownian motion corresponds to Hurst exponent equal to 0.5 and it is considered as the baseline uncorrelated process. Series with Hurst exponent greater than 0.5 are considered as persistent; series with Hurst exponent smaller than 0.5 are anti-persistent (consecutive values tend to be very different). Additionally, if the value of this exponent does not vary significantly with the magnitude of fluctuations, then the time series is considered monofractal and it can be consistently analysed via DFA; in the opposite case, it is multifractal and the MFDFA is a more suitable choice. If the studied time series corresponds to a sequence of discrete observables attached to the vertices of a network and the ordering is determined by the subsequent encounters of a random walker exploring the graph, then its persistence / antipersistence property can be translated into the assortative / disassortative character of the graph with respect to said observables. An assortative graph [131] is a graph in which vertices with similar properties (typically the degree is used, but in theory any property of the vertex can be taken into account) tend to be in contact more frequently than what expected by chance, while a disassortative graph has the opposite feature. Studying a complex network by the action of a random walker producing a collection of time series of encounters with vertices has an advantage with respect to the simple computation of the static assortative indexes of the graph. Indeed, the walker trajectories offer also a sampling of the paths distribution in the graph. This distribution is affected by the whole set of mutual relations of vertices at different scales, which are not fully appreciable by a single static snapshot of the network by means of classical network invariants. In the same manner, we are able to gain an insight on the different scaling of the autocorrelation function and hence on the distribution of the corresponding observable across different locations and scales of the network.

In this study, we primarily focus on the protein contact networks (PCN) elaborated from the E. coli [103]. We compare the properties of PCN with those of different known network and time series models. In particular, we bias the study on their analogies and differences with their synthetic counterpart PCN-S, introduced Sec. 1.6.1; PCN-S consist in coiled cords of polymers

in which the probability of contact is a decreasing function of the distance between residues along the chain.

3.1 Multi-Fractal Detrended Fluctuation Analysis

It is known that many processes in Nature and society present long-term memory, manifested in primis as heavy tails in the autocorrelation function of the considered observables. This phenomenon, referred to as the *persistence* of a process, can be characterized by the value of the Hurst exponent H, introduced in 1951 by the British hydrologist Harold Edwin Hurst [81]. The exponent normally assumes values in the range $[0,1]$ and is traditionally calculated with the R/S analysis, as shown in [153]. When the process corresponds to uncorrelated noise (e.g. Brownian motion) then the value of H is 0.5, whereas if the process is persistent (correlated) or antipersistent (anticorrelated) it will be respectively greater than and less than 0.5. However, conventional methods employed to analyze the long-range correlation properties of a time series (e.g., spectral analysis, Hurst analysis [21, 153]) reveal to be misleading when said time series is non-stationary. In fact, in many cases it is important to distinguish fluctuations caused by trending behaviors of data at all time scales – which in this context can be regarded as noise – from the intrinsic fluctuations characterizing the dynamical process generating the time series. One of the methods usually employed for this purpose is the Detrended Fluctuation Analysis (DFA), which has shown to be successful in a broad range of situations [139]. The DFA has been generalized in the so-called Multifractal Detrended Fluctuation Analysis (MFDFA) [23, 35, 94, 136], which accounts for multifractal scaling, that is, different correlation behaviors on different portions of data, which are thus identified by different sets of scaling exponents. Among the many applications of MFDFA, it is possible to cite the analysis of human EEG [179], solar magnetograms [113], human behavioral response [83], hippocampus signals [60], seismic series [158, 159], medical imaging [111], financial markets [22, 151], and written texts [10].

The MFDFA procedure is described thoroughly in [94] and it is reported briefly in the following. The method can be summarized in five steps, three of which are identical to the DFA version. Given a time series x_k of length N with compact support, the MFDFA steps are:

- *Step 1* : Compute $Y(i)$ as the cumulative sum (profile) of the series x_k:

$$Y(i) \equiv \sum_{k=1}^{i} [x_k - \langle x \rangle], \;\; i = 1, \ldots, N. \tag{3.1}$$

- *Step 2* : Divide $Y(i)$ in $N_s \equiv \text{int}(N/s)$ non-overlapping segments of equal length s. Since the series length N may not be a multiple of s, the last segment is likely to be shorter, so this operation is repeated in reverse order by starting from the opposite end of the series, thus obtaining a total of $2N_s$ segments.

- *Step 3* : Execute the local detrending operation by a suitable polynomial fitting on each of the $2N_s$ segments. Then determine the variance,

$$F^2(\nu, s) \equiv \frac{1}{s} \sum_{i=1}^{s} \left\{ Y[(\nu - 1)s + i] - y_\nu(i) \right\}^2, \tag{3.2}$$

  for each segment $\nu = 1, \ldots, N_s$ and

$$F^2(\nu, s) \equiv \sum_{i=1}^{s} \left\{ Y[N - (\nu - N_s)s + i] - y_\nu(i) \right\}^2 \tag{3.3}$$

for $\nu = N_s + 1, \ldots, 2N_s$, where $y_\nu(i)$ is the fitted polynomial in segment ν. The order m of the fitting polynomial, $y_\nu(i)$, determines the capability of the (MF-)DFA in eliminating trends in the series, thus it has to be tuned according to the expected maximum trending order of the time series.

- *Step 4* : Compute the qth-order average of the variance over all segments,

$$F_q(s) \equiv \left\{ \frac{1}{2N_s} \sum_{\nu=1}^{2N_s} \left[F^2(\nu, s)\right]^{q/2} \right\}^{1/q}, \tag{3.4}$$

with $q \in \mathbb{R}$. The q-dependence of the fluctuations function $F_q(s)$ highlights the contribution of fluctuations at different orders of magnitude. For $q > 0$ only the larger fluctuations contribute mostly to the average in Eq. 3.4; conversely, for $q < 0$ the magnitude of the smaller fluctuations is enhanced. For $q = 2$ the standard DFA procedure is obtained. The case $q = 0$ cannot be computed with the averaging form in Eq. 3.4 and so a logarithmic form has to be employed,

$$F_0(s) = \exp\left\{ \frac{1}{2N_s} \sum_{\nu=1}^{2N_s} \ln\left[F^2(\nu, s)\right] \right\}. \tag{3.5}$$

Steps 2 to 4 are repeated for different time scales s, where all values of s have to be chosen such that $s \geqslant m + 2$ to allow for a meaningful fitting of data. It is also convenient to avoid scales $s > N/4$ because of the statistical unreliability of such small numbers N_s of segments considered.

- *Step 5* : Determine the scaling behavior of the fluctuation functions by analyzing log-log plots of $F_q(s)$ versus s for each value of q. If the series x_i is long-range power-law correlated, $F_q(s)$ is approximated (for large values of s) by the form

$$F_q(s) \sim s^{h(q)}. \tag{3.6}$$

The exponent $h(q)$ is the *generalized Hurst exponent*; for $q = 2$ and stationary time series, $h(q)$ reduces to the standard Hurst exponent, H. When the time series manifests a uniform scaling over all magnitudes of fluctuations - i.e. $h(q)$ is independent of q - the series is said *monofractal*. Conversely, when different scaling behaviors are observed depending on q and $h(q)$ actively depends on q, the series is referred to as *multifractal*.

Starting from Eq. 3.4 and using Eq. 3.6, it is straightforward to obtain

$$\sum_{\nu=1}^{N/s} [F(\nu, s)]^q \sim s^{qh(q)-1}, \tag{3.7}$$

where, for simplicity, it has been assumed that the length N of the series is a multiple of the scale s, such that $N_s = N/s$. The exponent

$$\tau(q) = qh(q) - 1 \tag{3.8}$$

corresponds to the multifractal generalization of the fractal *mass exponent*. In case of positive stationary and normalized time series, $\tau(q)$ corresponds to the scaling exponent of the q-order partition function $Z_q(s)$. Another function that characterizes the multifractality of a series is the singularity spectrum, $D(\alpha)$, which is obtained via the Legendre transform of $\tau(q)$,

$$D(\alpha) = q\alpha - \tau(q), \tag{3.9}$$

where α is equal to the derivative $\tau'(q)$ and corresponds to the *Hölder exponent* (also called singularity exponent). Using Eq. 3.8 it is possible to directly relate α and $D(\alpha)$ to $h(q)$, obtaining:

$$\alpha = h(q) + qh'(q) \text{ and } D(\alpha) = q[\alpha - h(q)] + 1. \tag{3.10}$$

The *multifractal spectrum* in Eq. 3.9 allows to infer important information regarding the "degree of multifractality" and the specific sensitivity of the time series to fluctuations of different magnitudes. In fact, the width of the support of $D(\cdot)$ is an important quantitative indicator of the multifractal character of the series (the larger, the more multifractal a series is). Also the codomain of $D(\cdot)$ encodes useful information, since it corresponds to the dimension of the subset of the times series domain which is characterized by the singularity exponent α.

3.2 The Considered Data

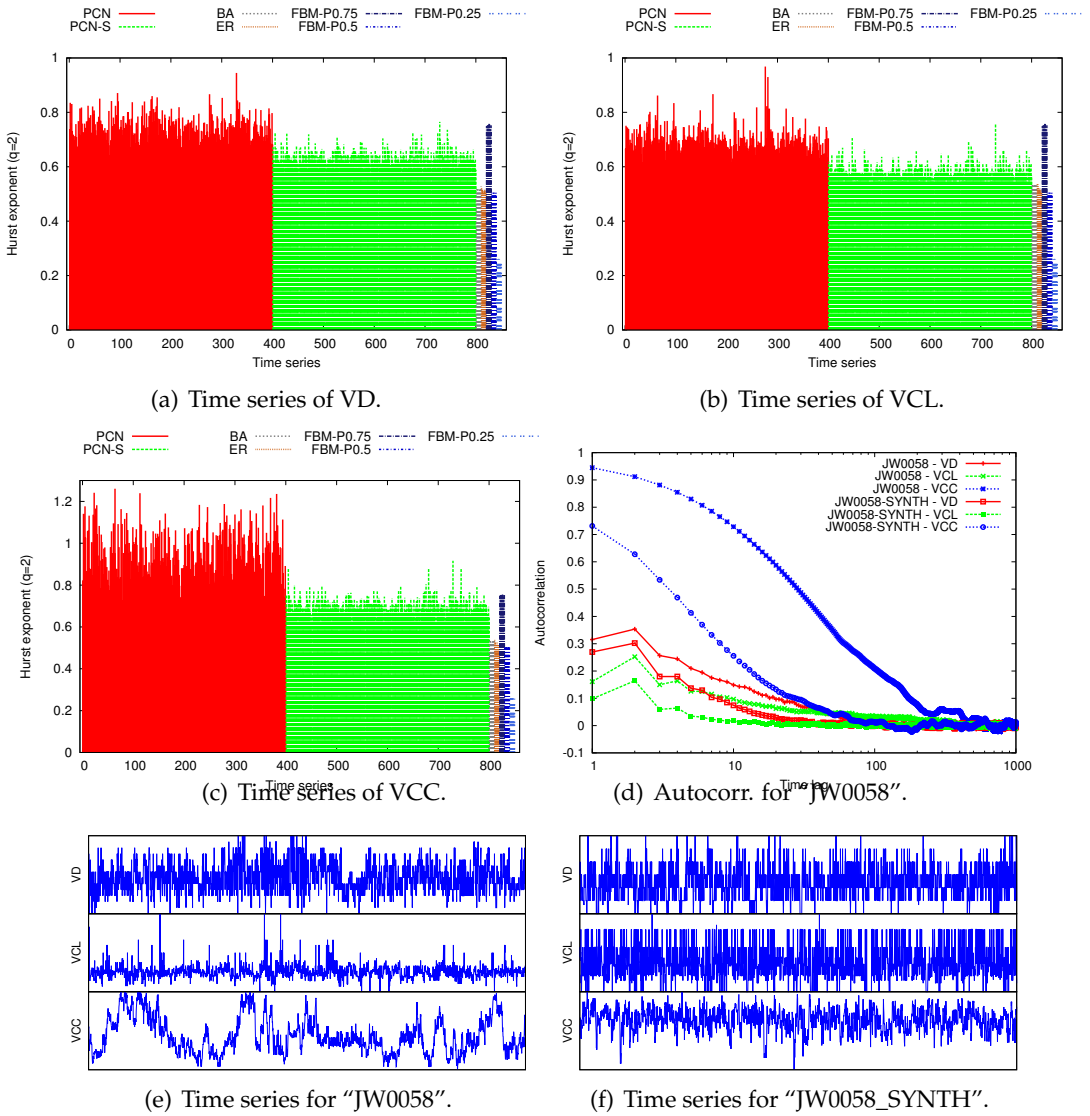
In this work we consider 400 E. coli protein contact networks (PCN) as the main object of study and we compare them to several models. To this end, we generated 400 synthetic polymers (PCN-S) by employing the generation method presented by Bartoli et al. [20], and by setting appropriate parameters in order to resemble the basic properties of each of the above PCN (i.e., the graph size). More precisely, each E. coli protein `JWxxxx` is juxtaposed with its synthetic counterpart, `JWxxxx_SYNTH`, having equal number of vertices and edges – the four-digit number `xxxx` stands for its unique identifier. In addition, we consider 10 Erdős-Rényi networks (ER) and 10 scale-free networks generated using the Barabási-Albert (BA) model, varying the number of vertices between 300 and 1200. The former are generated setting $p = \log(N)/N$, where N is the network size, while for the latter we used a six-degree attachment scheme. To allow the processing of such networks via the MFDFA procedure, we generate time series by means of stationary unbiased RWs, where at each step an observable is measured from the current vertex. Considering the size of the networks at hand, the RW length has been fixed at 10^5 time instants; this length assures the coverage of all vertices for a statistically significant number of times and it is consistent with the recommendations in Ref. [134]. We associate to each network three time series generated within the same RW. The first series considers vertex degree (VD) as observable; the second one the vertex clustering coefficient (VCL); the third one the vertex closeness centrality (VCL). Those three observables account for, respectively, the short, medium, and long range information of the network from the point of view of a vertex.

The dataset is also composed by six classes of time series that act as probes, which are obtained directly from their generative models. The herein considered time series are obtained from three fractional Brownian motion (FBM) processes with increasing Hurst coefficients, and three multifractal binomial cascades (BC), characterized by increasing MFS widths. FBMs have coefficients $H = 0.25, 0.5, 0.75$ and represent the poles of monofractality with increasing persistence. For each fixed value of H, we generated ten different time series (for a total of 30 FBMs) to account for the statistical variability. On the other hand, BCs are deterministic multiplicative processes, which are generated with the partition coefficient $a = 0.6, 0.7, 0.8$. These series are inherently multifractal, although they possess different persistence levels. Notice that in this case there is no point in generating more than one instance of the BC processes for each value of a, since the process is deterministic; so only three BC time series are generated.

3.3 Analysis of persistence properties

The first property that we analyze is the Hurst coefficient that, as described above, quantifies the persistence of the time series. In Figs. 3.1(a), 3.1(b), and 3.1(c) are shown the values of H

measured on each time series of the PCN, PCN-S, BA, and ER, for each of the three observables VD, VCL, and VCC, respectively, along with the Hurst exponents proper of the three classes of FBMs. Notice that, since FBM time series are not obtained as different observables yielded by a RW on a network, their Hurst exponents have been just replicated across the three figures.



(a) Time series of VD.

(b) Time series of VCL.

(c) Time series of VCC.

(d) Autocorr. for "JW0058".

(e) Time series for "JW0058".

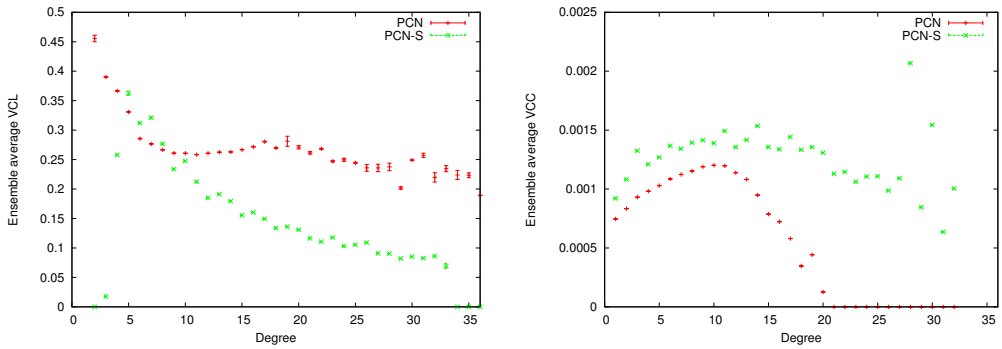
(f) Time series for "JW0058_SYNTH".

Figure 3.1. Persistence of the series measured through the Hurst exponent for VD (a), VCL (b), and VCC (c). The PCN (red bands) show significant persistence for all the three observables. (d) Sample autocorrelation function for the protein JW0058 and the corresponding synthetic polymer JW0058_SYNTH. (e) and (f) Sample time series. The higher persistence of the natural protein with respect to the synthetic analogue is particularly evident in the VCC series.

As expected, BA and ER networks produce RWs consistent with an uncorrelated Brownian motion (i.e., $H = 0.5$), since basically they are the result of an uncorrelated degree distribution. Interestingly, from the persistence levels shown in Fig. 3.1, it is possible to observe that PCN (red bands) induce time series with strong persistence, regardless of the particular observable. It is also evident from Fig. 3.1 that also synthetic polymers (green bands), similarly to the PCN, show positively correlated behaviours, even if they do not seem to capture this characteristic persistence to a sufficient degree. It is also important to mention that, when plotting the

Hurst exponents of PCN-S as a function of Hurst exponents of their corresponding PCN (data not shown for brevity), no trending has been observed. Indeed, these two quantities are not proportional and thus PCN-S instances cannot be considered just as less-persistent versions of their corresponding PCN. These results can be exploited to gain a more insightful view on the intrinsic characteristics of the PCN class, by relating the properties of the RWs to the topological properties of the corresponding graphs. In particular, the time series of VD show positive correlations, which in turn imply degree assortativity. This result is in agreement with the claims of Böde et al. [29], although we reached the same result by exploiting a different technique, since usually the degree assortativity is investigated through the method proposed by Newman [131]. It is worth pointing out that, since PCN are known to be fractal networks (embedded into a three-dimensional space) [68, 104], the observed degree assortativity is not in agreement with the theoretical hypothesis of Song et al. [156], which requires the degree distribution of fractal networks to be disassortative.

The high persistence of the clustering coefficient observed in the PCN is slightly more tricky to interpret in terms of topological properties. Roughly speaking, the VCL of a vertex is proportional to the local connectivity of the subgraph formed by the vertex and its closest neighbors with respect to the whole graph. It is known that PCN show a high degree of global modularity (see [50, 105]). Therefore, the persistence of the clustering coefficient can be interpreted as the tendency of vertices in the same module to be connected rather uniformly with the presence of medium-to-small hubs. As a confirmation of this fact, PCN do not have large hubs [29, 50]. Another way to explain this property is to directly relate VCL to the persistence of VD time series. To this end, Fig. 3.2(a) shows the relation of the degree-dependent average VCL over the possible VD. Here we considered the whole PCN and PCN-S ensembles. The error bars (which are usually smaller than the marker) represent the standard deviation over the entire ensemble. As it is possible to observe, while the two trends are substantially different, the standard deviation is very small in both cases for most values of the degree. This fact, along with the aforementioned degree assortativity, suggests a possible explanation for the persistence displayed by the VCL series of both PCN and PCN-S. It is worth noting that, for PCN, the clustering coefficient remains high when increasing the degree, which can be interpreted as a sign of high global modularity.



(a) VCL vs VD. (b) VCC vs VD.

Figure 3.2. Ensemble average VCL (a) and VCC (b) as a function of the degree for all proteins (red points) and synthetic counterpart (green points). Relation with respect to the degree shows significant difference among the real proteins and the synthetic polymers.

While VD and VCL show similar characteristics, the behaviour of the VCC observable is considerably different. By looking at the plot in Fig. 3.1(c), it is possible to observe that the Hurst coefficients of PCN are comparable and occasionally greater than one – i.e., the corresponding

time series are non-stationary. This might be considered as a symptom of different distributions of typical paths within the PCN. This conjecture would be in line with the observation of Yan et al. [173], where it is hypothesized that there are two characteristic distributions of paths within PCN, intra-module and inter-module paths, which is also a consequence of the PCN's high degree of modularity mentioned before. On the other hand, PCN-S do not share this feature, confirming an intrinsically different configuration of the network topology at a global scale. As for the VCL observable, the VCC persistence can be related to the VD persistence by inspecting Fig. 3.2(b). By comparing the two trends, it can also be observed that the PCN-S have a broader distribution over the possible degree values, as a consequence of being small-world networks [20], while PCN are neither small-world nor scale-free [50, 105]. In Fig. 3.1(d) it is shown the autocorrelation function of the three time series for one randomly chosen protein and its synthetic counterpart. First, it is worth noting that long-range correlations appear here in the form of heavy tails. Additionally, the VCC autocorrelation function denotes a much heavier tail with respect to the other observables, which is justified by the higher persistence (see Fig. 3.1(c)). To conclude, in Fig. 3.1(e) and 3.1(f) are shown two excerpts of the time series generated by the same protein and its synthetic model for each observable. By visually comparing the two plots, in particular for the observables VCL and VCC, it is possible to notice that the two networks generate RWs that are significantly different; the higher persistency of the PCN observables is also visually recognizable.

From these results it is clear that the PCN-S network models present significant discrepancies from their real counterparts, while still being distinguishable from other network models. These differences will be further analyzed in the following subsections.

3.3.1 Analysis of multifractal properties

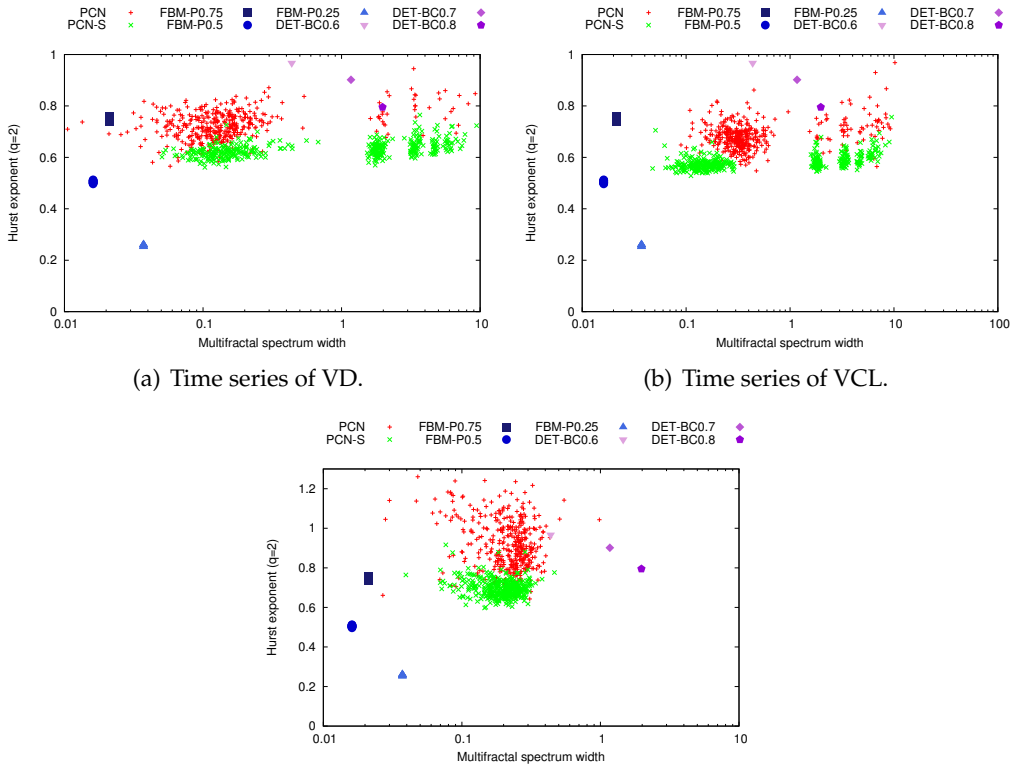
After having calculated the persistence properties of the considered time series, we can now proceed to evaluate their degree of multifractality. For each of the time series presented in Sec. 3.2, we perform the MFDFA procedure exposed in Sec. 3.1 by executing the Matlab® routine `MFDFA1()`, written by Ihlen and described in detail in Ref. [82]. The input of the routine is the time series to analyze, a vector of the considered time scales (corresponding to the set of increasing length scales s described in Sec. 3.2), the range of q-orders to be considered for the analysis, and finally the polynomial order, m, for the detrending. For the analysis of all time series, we used the following setting:

- the time scales $s \in \{16, 32, 64, 128, 256, 512, 1024\}$;
- the orders $q \in \{-5, -4.8, -4.6, \ldots, +4.8, +5\}$ for a total of 51 values;
- the detrending order $m = 2$.

The output produced by the routine, for all values of q, is the collection of (generalized) Hurst coefficients $H(q)$, mass exponents $\tau(q)$, singularity exponents $\alpha(q)$, dimension coefficients $D(\alpha(q))$, and scaling function $F(q)$. Please note that since $D(\alpha(q))$ is returned by the procedure directly as a function of q, in the following we will denote $D(\alpha(q))$ as $D(q)$. The width of the MFS is the extent of the $D(\cdot)$ support, which characterizes the degree of multifractality of a series. Clearly, all these quantities are not independent with each other and thus, in order to reduce redundancies, we only considered the subset consisting of $\alpha(q)$ and $D(q)$ in the embedding discussed later in Sec. 2.2. In fact, as said before, the MFS, $D(\cdot)$, encodes all information regarding the multifractality of the time series. Notice that all the networks are described simultaneously by three time series, corresponding to the three observables VD, VCL, and VCC,

while the probe time series are expressed by the same realization of the process for all the three observables.

To gain a first insight on the multifractality of the considered time series, it is useful to relate this property to the persistence levels calculated in Sec. 3.3. In particular, we perform this analysis for the PCN and the PCN-S since they exhibit the highest values of H; we consider here also the six probes, i.e., the three FBMs and three deterministic BC. Fig. 3.3(a), 3.3(b), and 3.3(c) show the plots of H versus the width of the MFS, respectively for the observables VD, VCL, and VCC.



(a) Time series of VD.

(b) Time series of VCL.

(c) Time series of VCC.

Figure 3.3. Hurst exponent vs MFS width of PCN and PCN-S. Both PCN and PCN-S show characteristics in-between mono and multi- fractal signals. Notice that the plot scale is log-lin for sake of clarity.

By comparing in Fig. 3.3 the relative distances between the PCN points and the probes, it is possible to observe that most PCN exhibit MFS widths that could be considered in-between those of mono and multi-fractal signals. Some proteins also show extremely wide MFS, while keeping the Hurst coefficient unaltered. Interestingly, PCN-S, while being less persistent, have a similar distribution of MFS widths. As observed for the persistence analysis, the VD and VCL observables behave very similarly also in terms of multifractality, while the VCC data points are more clustered and present slightly narrower spectra. Once again, this can be attributed to the substantial difference between the types of observables. Indeed, VD and VCL are short/medium range observables, so they can be influenced by the vertex position within the network at many distance scales. Instead, the VCC is mainly influenced by large scales (being a global topological descriptor), hence explaining why it shows less multifractal behaviour.

The variety of MFS herein observed justifies the experiments performed in the next section, which are focused on the analysis of the MFS projected in a suitable PCA space.

3.3.2 Embedding of the multifractal spectra

As mentioned above, the MFS elaborated from the time series constitutes the principal hallmark of all multifractal features. However, as first observed in Sec. 3.3.1, the spectra widths vary significantly even between members of the same class. Hence, there is no element that can be accounted for a meaningful representative of the whole class of proteins. For this reason, we embed all considered MFS coefficients, i.e., $D(q)$ and $\alpha(q)$, in a suitable low-dimensional vector space derived by means of a PCA. With the embedding into a PCA space, we are enabled to study the ensemble properties of each class without focusing on single elements alone, hence gaining an insight on the features that mostly characterize the particular typology of networks. In such embedding, each time series is initially represented by a vector $v \in \mathbb{R}^n$. Here, $n = 300$ is the total number of coefficients retrieved by the MFDFA that we consider, which is composed by 50 values of $D(q)$ plus 50 values of $\alpha(q)$, for each of the three observables. A given network G, associated with time series $x_G^{\mathrm{VD}}(t)$, $x_G^{\mathrm{VCL}}(t)$, and $x_G^{\mathrm{VCC}}(t)$, is thus represented by a vector $\vec{v}_G \in \mathbb{R}^{300}$ with the form

$$\begin{aligned}\vec{v}_G = \Big[& D_{\mathrm{VD}}(-5), \dots, D_{\mathrm{VD}}(+5),\ \alpha_{\mathrm{VD}}(-5), \dots, \alpha_{\mathrm{VD}}(+5), \\ & \hookrightarrow\ D_{\mathrm{VCL}}(-5), \dots, D_{\mathrm{VCL}}(+5),\ \alpha_{\mathrm{VCL}}(-5), \dots, \alpha_{\mathrm{VCL}}(+5), \\ & \hookrightarrow\ D_{\mathrm{VCC}}(-5), \dots, D_{\mathrm{VCC}}(+5),\ \alpha_{\mathrm{VCC}}(-5), \dots, \alpha_{\mathrm{VCC}}(+5) \Big]^{\top},\end{aligned} \tag{3.11}$$

where $D_{\mathscr{O}}(q)$ and $\alpha_{\mathscr{O}}(q)$, with $\mathscr{O} \in \{\mathrm{VD}, \mathrm{VCL}, \mathrm{VCC}\}$, are respectively the dimension coefficient and the singularity coefficient associated to the time series $x_G^{\mathscr{O}}(t)$ as a function of the order parameter q. We stress that in our analysis q assumes 51 equally-spaced values between -5 and 5, with a step size of 0.2; however, we do not consider the $q = 0$ case since it yields trivial values for the MFS.

On the other hand, the probe time series (FBM and BC) are not derived from a network. To be consistent with the aforementioned vector representation, their MFDFA coefficients are simply replicated 3 times, giving a vector of the form:

$$\begin{aligned}\vec{v}_{\mathrm{probe}} = \Big[& D(-5), \dots, D(+5),\ \alpha(-5), \dots, \alpha(+5), \\ & \hookrightarrow\ D(-5), \dots, D(+5),\ \alpha(-5), \dots, \alpha(+5), \\ & \hookrightarrow\ D(-5), \dots, D(+5),\ \alpha(-5), \dots, \alpha(+5) \Big]^{\top}.\end{aligned} \tag{3.12}$$

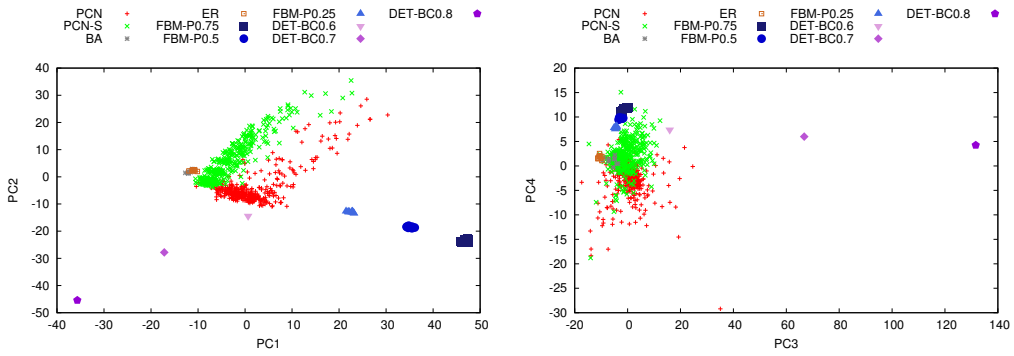
The 300-dimensional vector space described above is obviously unmanageable from the point of view of interpretation and, of course, visualization. For this reason, we perform a PCA to obtain a more synthetic description of the data. The PCA does not only allow to reduce the dimensionality of the data, but it also allows to give a reasonable and more direct interpretation of the new reference framework, i.e., the PCs. This is the main reason why we opted for PCA instead of a more sophisticated, non-linear, dimensionality reduction technique. Notice also that the process has been operated on the standardized data (z-scores), which corresponds to the correlation-based PCA, instead of the covariance-based version.

As shown in Tab. 3.1, the first four PCs explain more than the 83% of the entire variance. For this reason, we will move our analysis to the considerably simpler four-dimensional space spanned by the first four PCs, which are shown in the plots of Fig. 3.4; we consider the two-dimensional subspaces derived by PC1–PC2 (3.4(a)) and P3–PC4 (3.4(b)), respectively.

To understand the meaning of the PCs just retrieved, we analyze their loadings. In Fig. 3.5 are shown the correlations of each original variable with the first four PCs, where the original variables are ordered as described in Eqs. 3.11 and 3.12. As it is possible to note in Tab. 3.1,

Table 3.1. Explained variance of the first five PCs.

	PC1	PC2	PC3	PC4	PC5
Explained Variance(%)	31.12	29.46	16.58	6.71	4.94

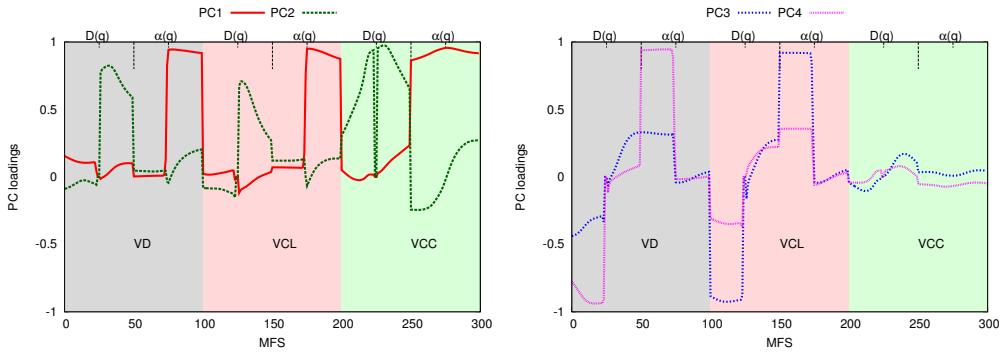


(a) PC1–PC2 (b) PC3–PC4

Figure 3.4. PCA of the MFS extracted from all considered time sseries.

the first two principal components, PC1 and PC2, are nearly equivalent in terms of explained variance and they are correlated, respectively, to the singularity exponent $\alpha(q)$ and spectrum $D(q)$. In particular, they are both strictly related to the large fluctuations (positive q orders) of the observables VD and VCL, and to almost all fluctuation orders of VCC – see Fig. 3.5(a). Once again, there is a clear separation between the characteristics of short and medium range observables, VD and VCL, and the long-range observable VCC. In fact, the discriminating power of VD and VCL is limited to the structure of their larger fluctuations, which is due to their local nature (we stress that local here refers to the neighborhood extent of the corresponding vertex). Arguably, the related small fluctuations behave just as a "background noise", providing little information on the relevant global properties of the networks. On the other hand, the organization of large fluctuations of VD and VCL in a RW indicates the occurrence and distribution of significant events, i.e., those related to the global topology of the network, like for example jumps between modules or areas with different local topology, hub encounters, etc. By following this interpretation, large fluctuations provide information that appears to play an important role in the discrimination of the network's class. The VCC, instead, is fundamentally different. In this case, as mentioned at the end of Sec. 3.3.1, the observable is much more sensitive since it is affected by the network topology at the largest distance scales. Hence, its variations are globally discriminating at all fluctuation orders.

The loadings of the third and fourth PC shown in Fig. 3.5(b) are easier to interpret. In fact, the first thing that is worth noting is the complete absence of influence of the VCC observable – since it is almost completely loaded in the first two PCs. This first observation reconfirms that all fluctuation orders of VCC provide important information in terms of variance. Interestingly, PC3 and PC4 are suitably allocated on the VCL and VD observables. At odds with what we have observed in Fig. 3.5(a), PC3 and PC4 are characteristic of the small fluctuations only (of both $D(q)$ and $\alpha(q)$). However, PC3 seems to be heavily influenced by the probe networks variance; Fig. 3.4(b) offers a visual understanding of this claim. Therefore, its contribution in the discrimination among the different network topologies is questionable, while PC4 provides a small but yet perceptible contribution in terms of variance.



(a) Loadings of PC1 and PC2 (b) Loadings of PC3 and PC4

Figure 3.5. PC loadings. The three observables forming the overall MFS are differentiated by using diverse shaded colors.

3.4 Discussion

In this chapter, we have exploited the possibility to characterize protein contact networks by means of a multifractal analysis of suitable time series. Such time series have been generated by performing stationary, unbiased, random walks on the graph structures, recording at each vertex three different quantities: the degree, clustering coefficient, and closeness centrality of the vertex. Those three observables capture, respectively, short, medium, and long range peculiarities of the considered networks. Our analysis of the considered protein contact networks was compared with several probe data. Notably, we used a receipt to generate synthetic polymers designed to mimic random coiled cords, two well-known classes of random networks, and two models of time series embodying the archetypical monofractal and multifractal signals.

The presented study provided a number of results. First, persistence analysis of the time series showed that proteins, regardless of the considered vertex observable, generate strongly persistent signals. When considering the degree as the observable, this can be translated into assortativity of the degree distribution. This first result is confirmed by the recent literature, although, to our knowledge, we are the first to asses such a property by means of time series analysis. We also pointed out that this result is in contrast with the recent hypothesis requiring disassortativity in fractal networks [65, 156, 177]. We also found that the assortativity of other observables can be linked to the assortativity of the vertex degree, since the degree basically controls the behaviour of the RW and thus influences to some extent all other measurements. Then we moved to a first analysis inspecting the multifractal footprint proper of the considered time series. Results showed that time series associated to protein contact networks – again regardless of the observable – should be considered as signals in-between the typical mono and multi- fractal behavior. We further elaborated over those results by performing the interpretation of the entire multifractal spectrum via the embedding into a suitable vector space. Such a vector space has been derived by first associating each time series to a high-dimensional vector containing suitable samplings of the domain and codomain of the multifractal spectrum derived by the multifractal dentrended fluctuation analysis. Successively, we performed a principal component analysis, resulting in a four-dimensional vector space explaining large part of the original data variance. The principal component analysis allowed us to perform a detailed interpretation regarding the importance of different fluctuation orders by analysing their loadings on the principal components. Results showed that large (in magnitude) fluctuations of all observables are more important in terms of discrimination (variance) of the considered networks/time series. Along with these, small fluctuations of the closeness centrality observable

were also recognized to be discriminating, fact that has been attributed to their long-range (global) nature. Small fluctuations of the degree and clustering coefficient, instead, are less informative, since they are more easily associated with background noise.

We conclude by arguing that this methodology for analyzing complex networks could be used also in different settings. Indeed, the techniques employed in this work never assume the knowledge of the global topology of the graph. In particular, this study might be of interest when the topology of the network under analysis is not directly observable, but can be gradually "explored" with suitable time-dependent measurements of the vertices. Moreover, the comparison of the proteins considered in this work with the corresponding synthetic versions highlighted important differences, which in turn strengthen the need to develop a more suitable generative model for protein contact networks. The problem of designing an improved generative model is discussed in the following Chapters.

Chapter 4

Detrending of time series with Echo State Networks

In the previous Chapter we discussed a methodology to indirectly gain insights on the topology of a network by studying the correlation properties of a random walk on its edges. To this aim, we employed the Multifractal Detrended Fluctuation Analysis, a procedure explained in Sec. 3.1 that measures the multifractal scaling properties of the fluctuations of a time series. In order to extract such quantities, the MFDFA performs a preprocessing step called *detrending*. Detrending is a fundamental step of the procedure since it allows to filter any nonstationary behavior in the time series, which would cause the appearance of spurious correlations in the results. The particular criterion chosen for the detrending is to divide the series in separate windows of a given size and fit a polynomial of a certain order in each window. The residuals are then considered as the detrended time series. However, trends are often present in the form of periodicities (also referred to as seasonalities) and/or fast-varying functions. These trends are not always correctly removed by the MFDFA procedure and this results in the detection of spurious correlation properties. For this reason, additional detrending methods are often used as a preprocessing step of the MFDFA in order to filter these trends before the polynomial detrending takes place. In other research works, the local detrending step of DFA is modified or replaced with other ad-hoc methods [79, 109, 144]. The main problem with detrending lies in the difficulty of defining what exactly a trend is [171]. Local-fit based methods rely on the assumption that a trend is generally a slow-varying process, while the superimposed noise is a process characterized by higher frequencies. While this is often the case, it is still difficult to determine the right form and parameters of the fitting function without biasing the analysis. Moreover, window-based fitting algorithms are heavily influenced by the choice of the window sizes. In [171] a trend is defined as an intrinsically fitted monotonic function or a function in which there can be at most one extremum within a given data span. This method is not affected by border effects since it is not window-based. However, a problem with this definition is that it does not (fully) describe periodic trends in a consistent way. Chianca et al. [36] suggested to perform a detrending by applying a simple low-pass filter, in order to eliminate slow periodic trends from data. While this approach is suitable for systems with slow-varying trends, it is difficult to apply to more general cases, when the trends' frequencies span over a significant portion of the (power) spectrum. Another approach that has been demonstrated to be useful in the case of periodicities was proposed by Nagarajan [129]. As a first step, the signal is represented as a matrix, whose dimension has to be much larger than the number of frequency components of the periodic (or quasi-periodic) trends as shown by the power spectrum. The well-known singular value decomposition method is then applied to remove components related to large-magnitude eigenvalues, which correspond to the trend.

Such a method, although interesting and mathematically well-founded, is very demanding in terms of computations and also assumes a deterministic form for trends.

In this Chapter, we follow an approach similar to Wu et al. [171] and define a trend in a completely data-driven way. We consider the analyzed time series as a series of noisy measurements of an unknown dynamical process. We also assume that the dynamical process is predictable to a certain degree by means of a particular type of Recurrent Neural Network (RNN) called Echo State Network (ESN) [25, 112]. RNNs have been shown to be able to predict the outcome of a number of dynamical processes [43]. In particular, a fundamental theorem formulated within the Neural Filtering framework, relates the number of neurons in a RNN hidden layer with the expected approximation accuracy of the estimated signal with respect to the true signal [110] of the process. Specifically, given a sufficiently large amount of processing units, a RNN that takes as input the measurement process can output an estimation that can be made as close as desired to the signal process, given its past input sequences. However, not all processes are predictable at the same level, as formally studied in [24, 41], for instance. For example, chaotic processes are not predictable for long time-steps, while other deterministic systems, like a sinusoidal waveform, can be easily predicted. In a stochastic setting, instead, we note that white noise cannot be predicted at all, since the past observations do not convey any information about the future. On the other hand correlated noise signals, such as fractional Gaussian noise (fGn), are in theory partially predictable given the presence of memory in the process. To handle prediction problems of increasing difficulty, models characterized by a higher complexity or a larger amount of training data are required. In the case of ESNs, the complexity of the model is mainly determined by the properties and the size of its recurrent hidden layer. Here we propose to perform a data-driven detrending of nonstationary, fractal and multifractal time series by using ESNs acting as a filter. In this study, trends are the only form of nonstationarities that we consider. By means of ESNs, we predict the trend of a given input time series, which is always superimposed to the (multi)fractal component of interest. Such a trend is then removed from the original time series and the residual signal is analyzed with MFDFA in order to evaluate its scaling and (multi)fractal properties. The proposed methodology is tested on several synthetic and real-world time series in order to assess its performance.

4.1 Detrending using ESNs

We now describe the main assumptions of our model and the detrending procedure to be used on a given univariate time series $y(t)$. We consider $y(t)$ as being composed of two superimposed components of different degrees of predictability:

- a trend process $x(t)$, which corresponds to the main stochastic process. This process represents the intrinsic dynamical evolution of the studied system and is predictable with high accuracy by an ESN;
- a noise process $n(t)$, which is less predictable by an ESN, hence requiring a more complex model to be described.

Under the assumption of statistical independence between $x(t)$ and $n(t)$, $y(t)$ can be separated in the sum

$$y(t) = x(t) + n(t), \quad t \in \mathbb{N}. \tag{4.1}$$

The trend $x(t)$ is a nonstationary stochastic process of larger magnitude with respect to $n(t)$, even if there are no hard constraints on their relative scales. The noise process, instead is a zero-mean, self-similar and stationary stochastic process which can in general be correlated, and

thus is characterized by a Hurst coefficient and a multifractal spectrum. Prototypical examples of such a process are fractional Gaussian noise and (fractional) Lévy stable processes [66, 148].

We are interested in removing the trend process from data and in obtaining the noise component $n(t)$ in order to be able to study its fractal properties. One way to approach this problem is to apply a filter to the measurement process and, in contrast with the common use of filters, only keep the noise part by subtracting the filtered signal from the original time series. A discrete-time optimal filter is a system that takes as input a measurement process $y(t)$ and outputs an estimate, $\bar{x}(t)$, of $x(t)$ at each time step t, such that a given error criterion (e.g., mean square error) is optimized. The simplest kind of filters are linear filters, which are widely employed in virtue of their efficiency and analytic tractability. However, in many situations not only the assumption of linearity is violated, but also an explicit analytical model of the signal is not available *a priori*. In these situations, it can be convenient to employ data-driven models that do not make strong assumptions on the data being processed and are capable to describe a wide range of processes.

In this work we employ an Echo State Network (see Appendix A) as a nonlinear filter in order to learn an approximation $\bar{x}(t)$ of the trend process $x(t)$, by training the system only with the measurement process $y(t)$. Since we are dealing with correlated noise, there is a possibility for an arbitrarily complex network to learn and predict also part of the noise process $n(t)$ and thus overfitting data. However, given our assumption of noise as a less predictable process, we constrain the neural network descriptive capability by using proper regularization techniques to prevent such overfitting. The proposed detrending with ESN procedure, called DESN, consists of a series of steps, whose details are provided in the following.

Let us consider the pair of time series $\{u_{\text{data}}(t), y_{\text{data}}(t)\}_{t=1}^{T}$ representing respectively the input and desired output of the network. Since in the prediction framework $y_{\text{data}}(t) = u_{\text{data}}(t+\tau_f)$, with τ_f the forecast horizon, the two time series can be constructed from a time series $\mathbf{z} = \{z(t)\}_{t=1}^{T+\tau_f}$, representing the measurements of the observed process. The two time series are then split into two separate datasets: training $\{u_{\text{tr}}(t), y_{\text{tr}}(t)\}_{t=1}^{T_{\text{tr}}}$ and test set $\{u_{\text{ts}}(t), y_{\text{ts}}(t)\}_{t=T_{\text{tr}}+1}^{T}$. The readout is trained by feeding the ESN with $u_{\text{tr}}(t)$ and forcing $y_{\text{tr}}(t)$ as teacher signal. At this point, the detrending procedure is applied on the remaining data of the test set. In particular, the prediction $\hat{y}_{\text{ts}}(t)$ is in turn utilized to detrend $y_{\text{ts}}(t)$, as explained below. From now on, we assume the ESN to be already trained and then, since the training data are no longer considered, we will denote $y_{\text{ts}}(t)$ simply as $y(t)$. The time series $\hat{y}(t)$, which denotes the values predicted by the ESN, can be expressed as:

$$\hat{y}(t) = y(t) + e_{\text{pred}}(t) = x(t) + n(t) + e_{\text{pred}}(t), \;\; t \in \mathbb{N}, \tag{4.2}$$

where $e_{\text{pred}}(t)$ is the ESN prediction error as a function of time.

The performance of a prediction model can be evaluated through the forecast accuracy, typically implemented as the normalized root mean square error [44], quantifying the differences between predicted and observed values. For a given model complexity, the prediction error is related to the amount of training data and on the accuracy of the training procedure. However, even for a optimally trained model, in the presence of noise the forecast will always be subject to an error, due to (intrinsic) stochastic unpredictability of the process or insufficient complexity of the prediction model. We refer to this source of error as *intrinsic unpredictability* of the process with respect to the given model complexity and its related error function as $e_{\text{intr}}(t)$. By assuming independence between the training error $e_{\text{tr}}(t)$ and the intrinsic error $e_{\text{intr}}(t)$, we can write $e_{\text{pred}}(t)$ as the sum of the independent components

$$e_{\text{pred}}(t) = e_{\text{tr}}(t) + e_{\text{intr}}(t), \;\; t \in \mathbb{N}. \tag{4.3}$$

If the prediction model is properly trained, we can assume the training error to be negligible, i.e.,

$$e_{\text{tr}}(t) \simeq 0 \; \forall t \in \mathbb{N}. \tag{4.4}$$

Our assumption in this work is that the trend process $x(t)$ of the observed signal $y(t)$ is completely predictable by an ESN model and all sources of intrinsic unpredictability are concentrated in the noise component $n(t)$. This assumption corresponds to approximating:

$$\hat{y}(t) = \bar{x}(t) \simeq x(t) \; \forall t \in \mathbb{N}. \tag{4.5}$$

When Eqs. (4.4) and (4.5) hold, by inserting Eq. (4.3) in (4.2) we obtain:

$$n(t) \simeq -e_{\text{intr}}(t) \; \forall t \in \mathbb{N}. \tag{4.6}$$

In this case, the predicted time series, $\hat{y}(t)$, is a good approximation $\bar{x}(t)$ of the trend component $x(t)$ of $y(t)$. Therefore, an estimation $\bar{n}(t)$ of the true noise $n(t)$ can be obtained as:

$$\bar{n}(t) \equiv y(t) - \hat{y}(t) = -e_{\text{pred}}(t) \simeq -e_{\text{intr}}(t). \tag{4.7}$$

The time series that we analyze here contains measurements of a signal with a superimposed noise, which increases the difficulty of obtaining high reliability in short-term forecasts. For this reason, one needs to wait until the trend accumulates sufficiently before it becomes clear: considering different forecast horizons could significantly influence the result of the prediction. In order to mitigate the dependency of the prediction performance on the particular forecast horizon τ_f, we perform multiple forecasts using an ensemble of k independent ESNs, each one trained considering a different prediction step-ahead $\tau_f^{(i)}, i = 1, ..., k$. The output signals of the ensemble of predictors, elaborated on the basis of the same input data but using different forecast horizons, generate independent outcomes $\hat{y}_i(t), i = 1, ..., k$, that are combined together in an average forecast, $\hat{y}(t) = 1/k \sum_{i=1}^{k} \hat{y}_i(t)$. This approach provides a more accurate prediction by compensating for the variance introduced by the single predictors. Such an approach is related to the well-known frameworks of ensemble learning [53, 160] and neural network ensembles [76]. In the latter it has been shown experimentally that the synergy of multiple back-propagation neural networks improved learning, generalization capability, noise tolerance, and self-organization with respect to a single, yet more complex system.

4.1.1 Other detrending methods

In this section, we describe some existing methodologies that have been used in previous works for separating trends from the noise components in a time series [23]. To be consistent with our approach, we consider the following detrending procedures as MFDFA preprocessing steps.

Empirical Mode Decomposition Empirical Mode Decomposition (EMD) is a data-driven technique that performs a decomposition of the original signal, $y(t)$, in terms of a finite number of modes $g_i(t)$, called Intrinsic Mode Functions (IMF), and a residual component. IMFs are derived directly from data, without any prior assumption about their model. EMD [61] can be used to extrapolate a trend in data by considering the residual given by: $\bar{x}(t) = y(t) - \sum_{i=1}^{n} g_i(t)$. The residue is hence subtracted from the original time series in order to remove the global trend and obtain an estimate of the noise. Generally, as shown in Wu et al. [171], also a number of IMFs are selected along the residual in order to better approximate the trend. This is especially needed where the trend is composed by periodicities, which cannot be approximated by a single residual. The EMD procedure has also been applied as a local detrending method in the windows computed with DFA, in place of the conventional polynomial fitting [144].

Fourier-Detrended Fluctuation Analysis The Fourier-Detrended Fluctuation Analysis (FDFA) is a tool used for identifying trends characterized by frequencies with a significant power [128]. The method targets the first few coefficients (those having larger amplitude or real part) of a Fourier expansion and thus it can be considered as a simple high-pass filter [36]. We use a slightly different approach here, which consists in cutting the spectral components with higher amplitude, rather than exclusively focusing on those having lower frequencies – as originally proposed in [36]. In this way, the definition of trends is relaxed in order to consider all larger amplitude periodicities, independently of their variation speed. Specifically, we first apply the discrete fast Fourier transform to the data records, then we sort the spectral components according to a decreasing order of their amplitude. Successively, we truncate the first τ_{freq} coefficients of the Fourier expansion. Finally, we apply the inverse Fourier transform to the truncated series. After this last step, *border effects* may appear at the opposite ends of the time series. These distortions are eliminated by cropping a portion of the initial and last part of the series.

Smoothing Smoothing methods operate in the time domain and basically implement low-pass filters. High frequency are attenuated on the base of the specific properties of the adopted smoothing method. We consider four different smoothing procedures, which depend on a parameter σ, representing the span of the smoothing procedure:

- Algorithm 1: a low-pass filter with coefficients equal to the reciprocal of the span (moving average);
- Algorithm 2: local regression using weighted linear least squares and a 1st degree polynomial model;
- Algorithm 3: local regression using weighted linear least squares and a 2nd degree polynomial model;
- Algorithm 4: a generalized moving average with filter coefficients determined by an unweighted linear least-squares regression and a polynomial model of specified degree p.

4.2 Experimental results

In this Section, we evaluate the performance of DESN, the proposed detrending method based on ESN. We compare the results with those obtained using the detrending methods introduced in Section 4.1.1, namely Empirical Mode Decomposition (EMD), Fourier-Detrended Fluctuation Analysis (FDFA), and different Smoothing (SM) techniques. In order to demonstrate the effectiveness of the proposed technique, we consider several synthetic time series having a self-similar noise component with known characteristics. We also test the methods on a real-world dataset, the sunspot time series, described in Section 4.2.2. These latter data have already been studied in the (multi)fractal analysis context – see, for example, [54, 79] and references therein. The datasets taken into account and a MATLAB code for reproducing all experiments presented in this Chapter are publicly available[1].

4.2.1 Synthetic time series

As described above, the synthetic time series are of the form $y(t) = x(t) + n(t)$, with $x(t)$ the trend and $n(t)$ the noise component. We use the four aforementioned detrending methods for

[1] `https://bitbucket.org/slackericida/desn_v1/overview`

computing an estimation of $x(t)$, namely $\bar{x}(t)$, and we evaluate the accuracy of each method by analyzing the LTC and multifractal properties of the estimated noise, $\bar{n}(t) = y(t) - \bar{x}(t)$. The accuracy of each method is evaluated by comparing the coefficients obtained with MFDFA (see Section 3.1) on the estimated noise $\bar{n}(t)$ with respect to the ground-truth $n(t)$. For all the synthetic series and methods, the MFDFA procedure has been executed on scales ranging from 16 to 1024 data points and with a second-order local polynomial detrending. The parameter q ranges from -5 to +5.

We consider seven time series `Y1`, ..., `Y7`, which are obtained by combining a trend selected from one of the five different time series `X1`, ..., `X5` with a noise selected from one of the three different time series `n1`, `n2`, and `n3`. Signals used as trend are described by the functions shown in Table 4.1. For the trend signals, `X1`,`X2`,`X4`, and `X5`, we report the interval from which the values of the domain variable x are extracted. In Table 4.2 are summarized the average properties of the synthetic noise components. We use two different sets of ten fGn processes generated by setting H respectively to 0.7 and 0.3, and a deterministic binomial multifractal cascade [136] with multiplicative factor equal to 0.60708. For the noise `n3`, we also consider the spectrum asymmetry

$$\Theta = \frac{\Delta\alpha_{\mathrm{L}} - \Delta\alpha_R}{\Delta\alpha_{\mathrm{L}} + \Delta\alpha_R}, \tag{4.8}$$

where $\Delta\alpha_{\mathrm{L}}$ and $\Delta\alpha_{\mathrm{R}}$ are the width of the left and right part of the support of $D(\alpha)$ (3.9), respectively. A negative value for Θ denotes a right-sided spectrum, highlighting a stronger multifractality on smaller fluctuations, while the contrary holds in the case of a positive value. All time series have been normalized by calculating the z-score; the amplitudes of signal and noise series are multiplied by a suitable scalar value, in order to obtain a signal-to-noise ratio of 16.

Table 4.1. Description of the functions used as trend within the synthetic signals. The term $\nu_{\max}$ refers to the Nyquist frequency $f_{\mathrm{s}}/2$, where f_{s} is the sampling rate, and the terms $\mathscr{U}(x_{\min}, x_{\max})$ and $\mathscr{N}(\mu_x, \sigma_x)$ are respectively the uniform and normal distributions.

ID	Description
`X1`	$\sin(t)$.
`X2`	$\sum_{i=1}^{10} A_i \sin(2\pi\nu_i t), \quad \{\nu_i = \mathscr{U}(0, 10^{-5}\nu_{\max})\}, \{A_i = \mathscr{N}(1,1)\}$.
`X3`	$s\|s\|s\|...$, with s the first 100 digits of π.
`X4`	$\sum_{i=1}^{10} A_i \sin(2\pi\nu_i t), \quad \{\nu_i = \mathscr{U}(0, 0.5\nu_{\max})\}, \{A_i = \mathscr{N}(1,1)\}$.
`X5`	$\sin(t)/t^2$.

Table 4.2. Characteristics of the synthetic noise processes. The Hurst exponent and MFW of `n1` and `n2` are the outcome of MFDFA averaged over ten independent realizations of the process.

ID	Description	Length	avg. Hurst	avg. MFW (Θ)
`n1`	fGn	150000	0.695	0.022
`n2`	fGn	150000	0.303	0.032
`n3`	Binomial cascade	131072	0.883	1.192 (0.048)

Overall, we performed seven different tests. In Table 4.3, we report the time series under consideration and the values used for configuring each detrending procedure. Note that the length of the i-th time series `Yi` is given by the length of the noise component, which is reported in Table 4.2. For DESN, we consider an additional time series for training the network (referred as $y_{\text{tr}}(t)$ in Section 4.1), whose length is half of `Yi`'s length.

Table 4.3. Time series and configuration of the different detrending procedures used in each test. For DESN, we report the values of the size of the reservoir (N_r), the spectral radius (ρ), the regularization coefficient (λ), and the number k of forecast models. For FDFA, we report the thresholds τ_{freq} and τ_{time} used for determining the amount of coefficients to be truncated in both frequency and time domain. For SM, we report the span of the moving average σ and the identifier of the adopted algorithm. Finally, for EMD we report the number of the last s IMFs which are used for defining the trend.

Data	DESN	FDFA	SM	EMD
`Y1 = X1 + n1`	$N_r = 500$, $\rho = 0.99$, $\lambda = 0.1$, $k = 30$	$\tau_{\text{freq}} = 150$, $\tau_{\text{time}} = 950$	$\sigma = 50$, algo: 2	$s = 13$
`Y2 = X2 + n1`	$N_r = 200$, $\rho = 0.4$, $\lambda = 0.1$, $k = 20$	$\tau_{\text{freq}} = 60$, $\tau_{\text{time}} = 1$	$\sigma = 1800$, algo: 3	$s = 5$
`Y3 = X3 + n1`	$N_r = 500$, $\rho = 0.99$, $\lambda = 0.1$, $k = 20$	$\tau_{\text{freq}} = 115$, $\tau_{\text{time}} = 50$	$\sigma = 20$, algo: 4	$s = 19$
`Y4 = X4 + n1`	$N_r = 400$, $\rho = 0.99$, $\lambda = 0.1$, $k = 10$	$\tau_{\text{freq}} = 400$, $\tau_{\text{time}} = 3000$	$\sigma = 10$, algo: 1	$s = 17$
`Y5 = X5 + n1`	$N_r = 100$, $\rho = 0.99$, $\lambda = 0.05$, $k = 30$	$\tau_{\text{freq}} = 4000$, $\tau_{\text{time}} = 250$	$\sigma = 1000$, algo: 1	$s = 8$
`Y6 = X1 + n2`	$N_r = 500$, $\rho = 0.99$, $\lambda = 0.1$, $k = 30$	$\tau_{\text{freq}} = 400$, $\tau_{\text{time}} = 2000$	$\sigma = 50$, algo: 2	$s = 17$
`Y7 = X1 + n3`	$N_r = 500$, $\rho = 0.99$, $\lambda = 0.05$, $k = 20$	$\tau_{\text{freq}} = 250$, $\tau_{\text{time}} = 2000$	$\sigma = 60$, algo: 4	$s = 24$

Results are obtained by averaging ten independent realizations of the tests. The sources of randomicity for each test are the different realizations of the noise process – for `n1` and `n2` – and the different executions of the DESN procedure – ESN input and reservoir weights. We used a grid search to tune the (hyper-)parameters of the different methods in their respective spaces. For each detrending method, we considered a different sets of bounds and search resolutions of the respective parameter space and a specific loss function for guiding the optimization. The error measurement that we used is the normalized root mean squared error (NRMSE) function, which is defined as follows:

$$\text{NRMSE} = \sqrt{\frac{\langle \|\mathbf{y} - \mathbf{d}\|^2 \rangle}{\langle \|\mathbf{y} - \langle \mathbf{d} \rangle\|^2 \rangle}}, \tag{4.9}$$

being $\mathbf{y}$ the ESN output (A.2) and $\mathbf{d}$ the desired one.

Parameter settings of detrending methods For DESN, the parameters that we considered are the size N_r of the reservoir, searched in $[100, 500]$ with resolution 100; the spectral radius ρ

searched in the set $\{0.4, 0.518, 0.636, 0.754, 0.872, 0.99\}$; the regularization coefficient λ used in the linear regression for the training of the readout is searched in $[0.05, 0.3]$ with step size 0.05; the number k of forecast models used is searched in $[10, 30]$ with step size 10. As discussed in Section 4.1, we used a different forecast step for training each of the k ESNs of the ensemble. In particular, the forecast step of the i-th predictor model is $m_k = 10 \cdot i$. The adopted loss function is the average error computed on y and forecast $\hat{y}_i$ of the i-th prediction model, that is, $1/k \sum_{i=1}^{k} \text{NRMSE}(\hat{y}_i, y)$.

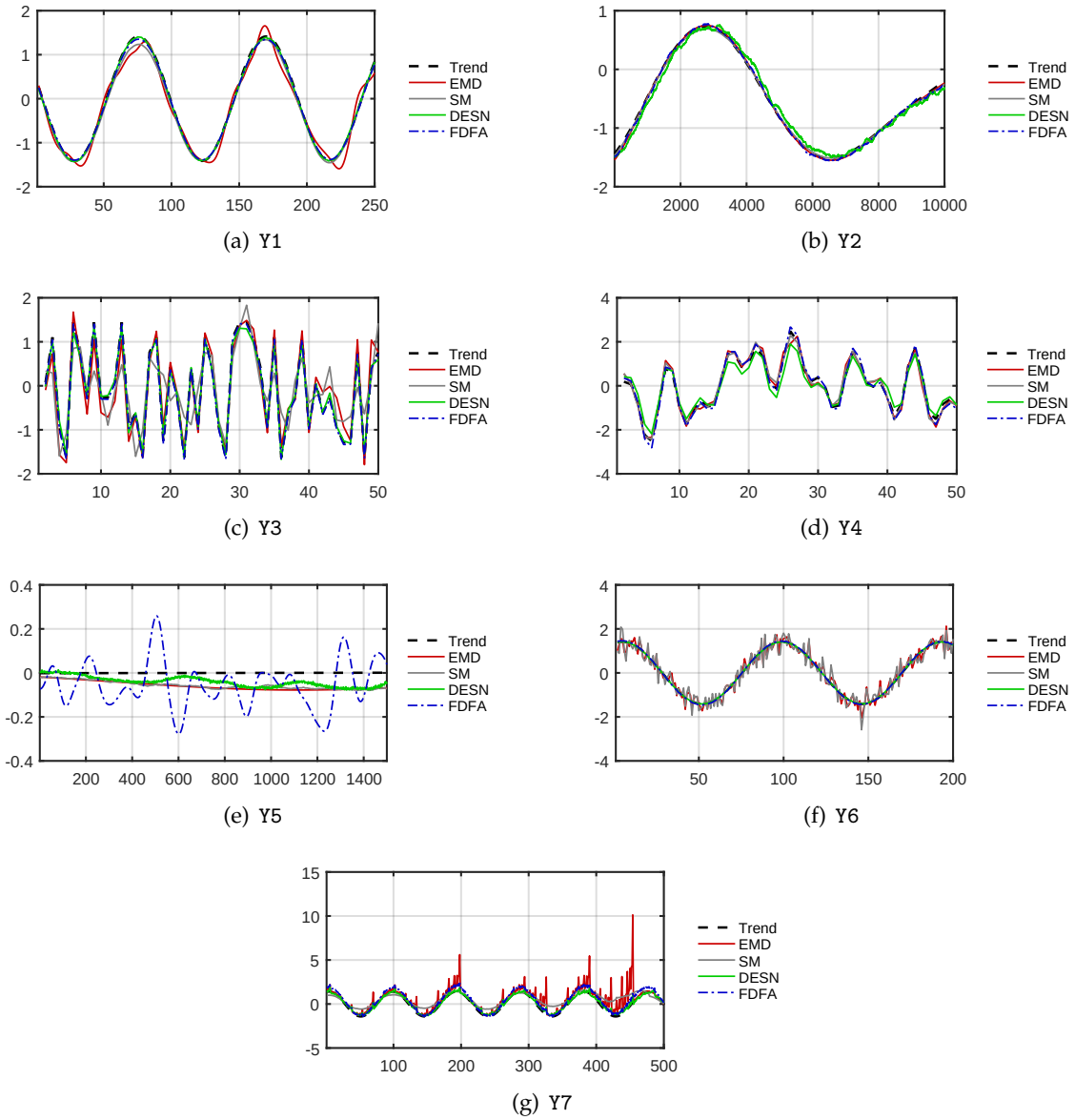
For the SM procedure, we tuned the span of the moving average σ in $[10, 2000]$ with step size 10. For guiding the hyper-parameter optimization, we used a loss function which minimizes the error and maximizes the span, defined as: $f_{\text{SM}} = \eta_{\text{SM}} \cdot \text{Err} + (1 - \eta_{\text{SM}}) 1/\sigma$, where Err is the error evaluated as Eq. (4.9) and $\eta_{\text{SM}} \in [0, 1]$ is a weight parameter that was set to 0.1 in every test. Note that for $\eta_{\text{SM}} = 0$ the error component is neglected, then the resulting span is maximized covering the whole time series; this generates a smooth function which assumes in every point the mean value of the original signal. On the other hand, by setting $\eta_{\text{SM}} = 1$, only the error is minimized and the span assumes its minimum value $\sigma = 2$, which generally produces an insufficient smoothing of the signal. We evaluated the performances using all the four algorithms described in Section 4.1.1 and we reported here the one which achieved the best results. The polynomial degree p in the algorithm 4 was set to 15 in every test.

For setting the optimal values of the parameter τ_{freq} in the FDFA procedure, after having ordered the Fourier coefficients by their amplitude (from larger to smaller), by visual inspection we first identify the "elbow" in the sequence, which is its inflection point, which determines the frequencies to be truncated (i.e., these having very high power). Once the inverse Fourier transform is performed, some cropping on the boundaries of the time series is necessary to attenuate boundary effects caused by the alteration of the spectrum.

Finally, in the EMD approach we used the standard setup of the stop criterion for retrieving the IMFs, as described in [80]. The sum of the last s IMFs represents the trend and the number s is optimized by minimizing the following loss function: $f_{\text{EMD}} = \eta_{\text{EMD}} \text{Err} + (1 - \eta_{\text{EMD}}) s/S$, where S represents the total number of IMFs identified relative to each signal – usually between 15 and 20 components. Also in this case, Err is the error evaluated with Eq. (4.9) and $\eta_{\text{EMD}} \in [0, 1]$ is a weight parameter. Note that for $\eta_{\text{EMD}} = 0$ the error component is neglected and s assumes the minimum value 1, i.e., only the last IMF is selected for approximating the trend. On the other hand, when $\eta_{\text{EMD}} = 1$ the error is minimized, but all the s IMFs are selected for representing the trend, which then coincides with the original signal. We set $\eta_{\text{EMD}} = 0.1$ when we tested the synthetic signals `Y3`, `Y4`, `Y6`, and `Y7`, while in the processing of the remaining signals (including the sunspot time series) we set $\eta_{\text{EMD}} = 0.5$.

Discussion of results In Fig. 4.1, we plot a short sample of each time series with superimposed the trends identified by the different detrending procedures. The details of the results are reported in Table 4.4, where we show the resulting Hurst coefficient and multifractal spectrum width (MFW) for each time series, with their corresponding standard deviations. In Fig. 4.2 we graphically represent the quality of the scaling of the fluctuation function for the estimated noise components. The linear fittings of the scaling functions are highlighted in green when the considered detrending method (column) has preserved a correct scaling behavior on the selected time series (row), while we used red dashed linear fittings to denote an incorrect scaling or significantly altered Hurst/MFW coefficients with respect to the ground truth.

As shown in Table 4.4, the four methods perform differently on each time series. With the EMD and SM methods, and considering the parameter optimization criteria presented in Section 4.1.1, we could not obtain a correct scaling for most of the tested time series. The first five time series, `Y1-5`, are composed by a signal (trend) with a superimposed persistent noise



(a) Y1 (b) Y2

(c) Y3 (d) Y4

(e) Y5 (f) Y6

(g) Y7

Figure 4.1. Colors online. Trends identified on the different signals. The function depicted with black dashed lines represents the trend of the original time series. The colored lines represent the trends identified using DESN, EMD, FDFA, and SM. For clarity of representation only small portions of the time series are shown.

with $H = 0.7$, according to Table 4.3.

In Y1, the trend is a single sinusoid, which is the simplest periodic function and it is easily separable from noise, which is much more complex from a prediction perspective. As expected, the Hurst exponent is estimated with a good precision by DESN. FDFA obtains a similar accuracy, since in this case the trend can be easily isolated, it being described by a single high-amplitude frequency in the Fourier domain. In fact, as described in Section 4.1.1, FDFA operates by eliminating the frequencies with largest amplitudes, so its maximum efficiency is reached when trends consists of few isolated dominating frequencies. On the other hand, in time series where trend periodicities are spread over a large portion of the spectrum or are too entwined with the noise frequencies, FDFA tends to fail. In fact, by cutting a significant amount

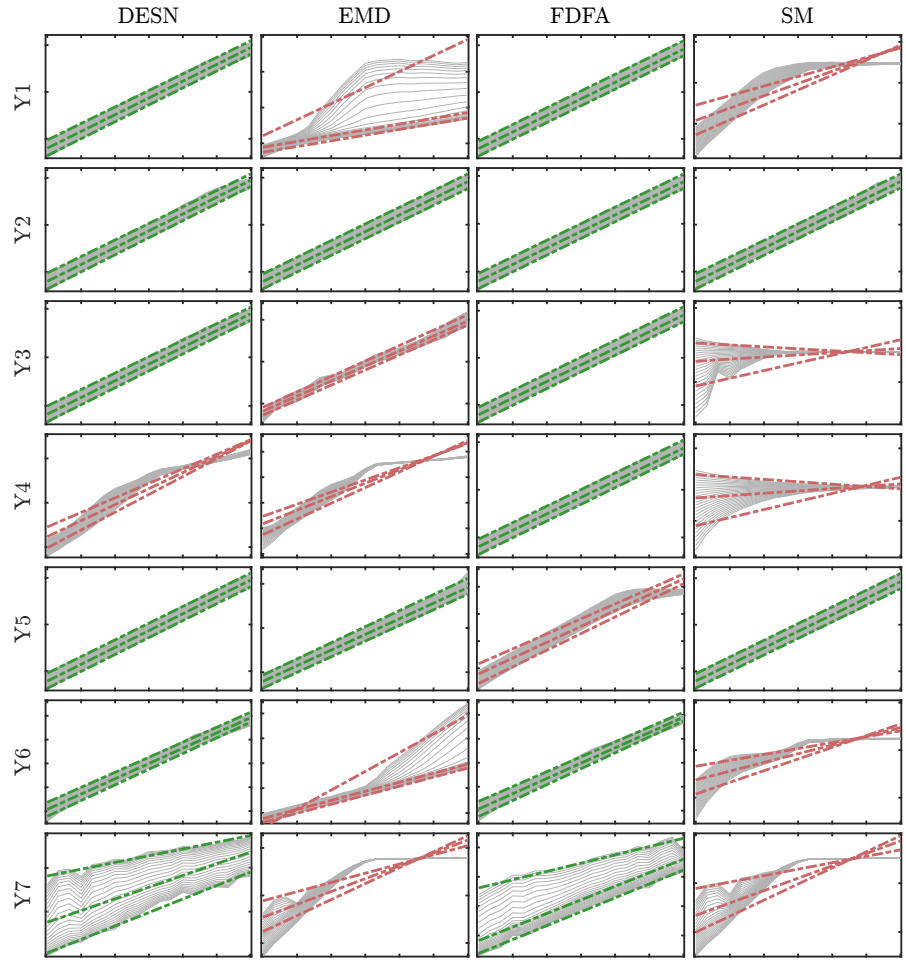


Figure 4.2. Colors online. Scaling of fluctuation functions related to the detrended time series. Only one instance of each test in Table 4.4 is reported here. The least-square linear fittings are highlighted in green when they correspond to a correct scaling function and in red otherwise.

of frequencies, FDFA tends to corrupt the spectrum of noise and hence its scaling properties. It is important to point out that the original FDFA method proposed in [36] works only as a low-pass filter without taking into account amplitudes, so its limitation is even more evident in these particular cases. The SM and EMD procedures do not perform well on identifying the trend in `Y1`. While this is a common issue with EMD applied to sinusoidal signals [171], with SM we can observe in the example of Fig. 4.2 a crossover that breaks the global scaling. This crossover is given by the smoothing algorithm acting only at a scale determined by its span parameter.

Despite the apparent increasing difficulty of the detrending task on the second time series `Y2`, whose trend is a linear combination of low-frequency sinusoids with different amplitudes, all methods perform equally well. However, by comparing the trend functions in Table 4.1, it is important to notice that the frequency of the sinusoid function in `X1` is significantly higher than the maximum value of the frequencies characterizing the trend `X2`. In this case, in fact, the variation of the trend signal is sufficiently slow to be isolated properly by EMD and SM, which behave in this case as low-pass filters.

On the third series `Y3`, the results are similar to what observed in the first test. In fact, the trend signal is a periodic series obtained by repeatedly concatenating the first 100 digits of π.

Therefore, the trend is characterized by a broad spectrum with fast frequencies, and thus EMD and SM are once again unable to perform the required task. In fact, even if from Fig. 4.2 we can observe the log-log scaling of EMD to be approximately linear, the obtained Hurst coefficient is 0.366, which differs significantly from the true value of 0.695 and incorrectly denoting an antipersistent behavior. This means that the fractal properties of noise have been considerably altered by the EMD detrending procedure and the result is not to be considered correct.

The trend in `Y4` is a more complex version of `Y2`, since the signal `X4` is characterized also by high frequencies, it being composed by a linear combination of 10 sine waves with frequencies chosen randomly in a broad interval. In this case, only FDFA succeeds in detrending the series correctly, since the spectrum of the trend consists of isolated high-amplitude frequencies. In fact, as explained above, the FDFA procedure implemented in this work filters the spectral components with greater amplitudes, regardless of their frequency, thus making the filtering method independent of the variation speed of the signal. EMD and SM, instead, are designed with the underlying assumption that trends are characterized by low frequencies (slow variation) and hence they are unable to filter rapidly-varying trends correctly. DESN, on the other hand, does not perform any explicit assumption regarding the form of the trend. In this case, however, the resulting signal is much harder to predict since its periodicity is much longer than the network's memory can account for. In particular, it has been shown that ESNs are unable to learn functions composed of even two superimposed oscillators with incommensurable frequencies [86], because of the aperiodicity of the compound signal. Such a signal, in fact, would require the simultaneous coexistence of two stable and uncoupled oscillating modes in the network's dynamics, a configuration that is very difficult to attain in practice.

The time series `Y5` is instead a classic example where the FDFA method fails. In this case, the trend signal does not consist of isolated frequencies, but it is described by a continuous distribution of frequencies in the spectrum, most of them characterized by a small amplitude. Hence, the filtering performed by FDFA alters the signal and this results in a crossover at larger scales, as we observe in Fig. 4.2. All the other methods, instead, perform well on this time series, given the regular behavior of its trend signal in the time domain and the prevalence of low frequencies in the Fourier domain.

The time series `Y6` is composed by the trend `X1` with the addition of antipersistent noise. Analogously as what observed for `Y1`, only DESN and FDFA succeed in correctly identifying the trend on such a time series. So far, in every test the estimation of `n1` and `n2` resulted to be monofractal, as confirmed by the estimated MFWs shown in Table 4.4. The only exception is in the outcome given by EMD on `Y5`, where we detect on $\bar{n}(t)$ the presence of spurious multifractal scaling, which is not present in the ground-truth signal `n1`.

The time series `Y7` is the only series characterized by a multifractal scaling. As shown in the results, in this case only DESN and FDFA produce a correct scaling function, even if the precision of the estimation is not optimal, probably because of the higher complexity of such a time series. The calculated Hurst coefficient is (slightly) overestimated by DESN and underestimated by FDFA. The principal difference in performance between these two approaches lies in the estimated multifractal spectrum width. In fact, in this case the estimate obtained with DESN is significantly closer to the ground truth, while FDFA considerably underestimates its value, thus suggesting a process with far less multifractal properties. Moreover, we can observe that both methods overestimate the asymmetry with a bias on the left-hand side of the spectrum. In the case of DESN, this can be explained by considering that the right-hand side of the spectrum corresponds to the smaller fluctuations, which are more easily affected by the ESN prediction error.

Table 4.4. Average values and standard deviations (where applicable) of Hurst exponent and width of the multifractal spectrum (MFW) of the noise estimated on each time series along with the ground truth (GT) value evaluated on the original noise. The asymmetry Θ of the multifractal spectrum (Eq. (4.8)) of the series `Y7` is reported in brackets. The standard deviation is not defined for the results of FDFA on series `Y7`, since the values are deterministic. The cases in which the detrending method did not succeed in preserving the noise self-similarity are denoted with "n.s.".

	ID	GT	DESN	FDFA	SM	EMD
Hurst	Y1	0.695	0.713 ± 0.007	0.705 ± 0.007	n.s.	n.s.
	Y2	0.695	0.719 ± 0.007	0.690 ± 0.004	0.706 ± 0.005	0.701 ± 0.006
	Y3	0.695	0.691 ± 0.006	0.702 ± 0.006	n.s.	0.366 ± 0.004
	Y4	0.695	n.s.	0.687 ± 0.002	n.s.	n.s.
	Y5	0.695	0.718 ± 0.006	n.s.	0.711 ± 0.006	0.711 ± 0.007
	Y6	0.303	0.318 ± 0.003	0.314 ± 0.002	n.s.	n.s.
	Y7	0.883	1.021 ± 0.003	0.793	n.s.	n.s.
MFW (Θ)	Y1	0.022	0.027 ± 0.012	0.026 ± 0.006	n.s.	n.s.
	Y2	0.022	0.032 ± 0.014	0.034 ± 0.013	0.028 ± 0.011	0.023 ± 0.009
	Y3	0.022	0.029 ± 0.008	0.024 ± 0.006	n.s.	0.023 ± 0.005
	Y4	0.022	n.s.	0.037 ± 0.010	n.s.	n.s.
	Y5	0.022	0.019 ± 0.007	n.s.	0.018 ± 0.005	0.102 ± 0.041
	Y6	0.032	0.040 ± 0.008	0.043 ± 0.002	n.s.	n.s.
	Y7	1.192 (0.048)	1.116 ± 0.046 (0.397 ± 0.060)	0.593 (0.849)	n.s.	n.s.

4.2.2 Sunspot data

In this section, we consider the time series relative to the number of daily sunspots [3]. The dataset contains more than 70000 records and is characterized by a trend given by the well-known 11-year cycle of the sun. Such a dataset has been already used by other authors in the field of (multi)fractal time series analysis (see, e.g., [54, 79]). For all the methods taken into account here, the MFDFA procedure has been executed on the detrended series with scale parameter ranging from 16 to 1024 data points, first-order local polynomial detrending, and parameter q ranging from -5 to +5.

For this test, we configured FDFA with $\tau_{\text{freq}} = 150$ and $\tau_{\text{time}} = 500$. In the EMD case, we set the weight parameter $\eta_{\text{EMD}} = 0.5$ in the cost function. For SM, we set the span $\sigma = 1000$, the weight parameter $\eta_{\text{SM}} = 0.1$, and we used algorithm 2. For DESN, we set the reservoir size $N_r = 500$, the regularization coefficient $\lambda = 0.05$, and the spectral radius $\rho = 0.99$. For DESN, we compared two settings with different numbers k of forecast models, namely $k = 10$ and $k = 30$, which produced slightly different, yet qualitatively comparable results. Since there is no known ground truth for the sunspot time series, in this section we compare our results with the properties reported in other works [54, 79].

In Table 4.5, we show the values of the Hurst coefficient and the width of the multifractal spectrum. As we can see in the table, all four methods, when suitably tuned, agree on the persistence of the process up to fluctuations of ~ 0.05 in the Hurst exponent values. Such values are also similar to the coefficient $H = 0.73$ reported in Ref. [79], where an adaptive detrending

is performed on the time series relative to monthly sunspot. The Hurst exponent retrieved with DESN, with an ensemble of $k = 10$ ESNs, is closer to the ground truth with respect to the other methods, while the outcome obtained with $k = 30$ is slightly higher. By assuming that the true value lies in-between the general consensus, this may suggest that a suitable dimension of the ESN ensemble has to be chosen in order to obtain best performance, even if the observed variability is in general fairly low. Regarding the MFW, we observe that DESN is not in agreement with the other detrending methods and, to a lower extent, also on the asymmetry Θ. In fact, even if all methods agree on the right-sided multifractal nature of the series, both DESN configurations denote a lower degree of multifractality and lower asymmetry. However, it is worth noting that the MFW value estimated by DESN is much closer to the values reported in [54], while the degree of asymmetry is still different. It is also worth pointing out that the authors in [54] did not perform any detrending in their work. This was possible thanks to the fact that the underlying trend is very slow and a number of sufficient data points can be analyzed by considering scales lower than half of the dominating periodicity. In Fig. 4.3, we show the trends identified using the different approaches herein taken into account. As it is possible to observe, the trend calculated by DESN correctly recognizes the characteristic 11-year cycle of the sunspot time series. In Fig. 4.4, we show the results of the scaling of the fluctuation function obtained by using the two configurations for k of DESN. The general agreement of the values estimated by DESN with other methods offers a sound justification for the quality and reliability of the proposed detrending method.

Table 4.5. Hurst exponent, MFW, and asymmetry (Θ) of the detrended sunspot time series, estimated using different detrending methods.

Method	**Hurst**	**MFW**	■
DESN ($k = 10$)	0.729 ± 0.0003	0.456 ± 0.0560	-0.408 ± 0.0536
DESN ($k = 30$)	0.808 ± 0.0002	0.641 ± 0.0614	-0.542 ± 0.0412
FDFA	0.688	1.205	-0.556
SM	0.680	1.118	-0.726
EMD	0.731	1.686	-0.786

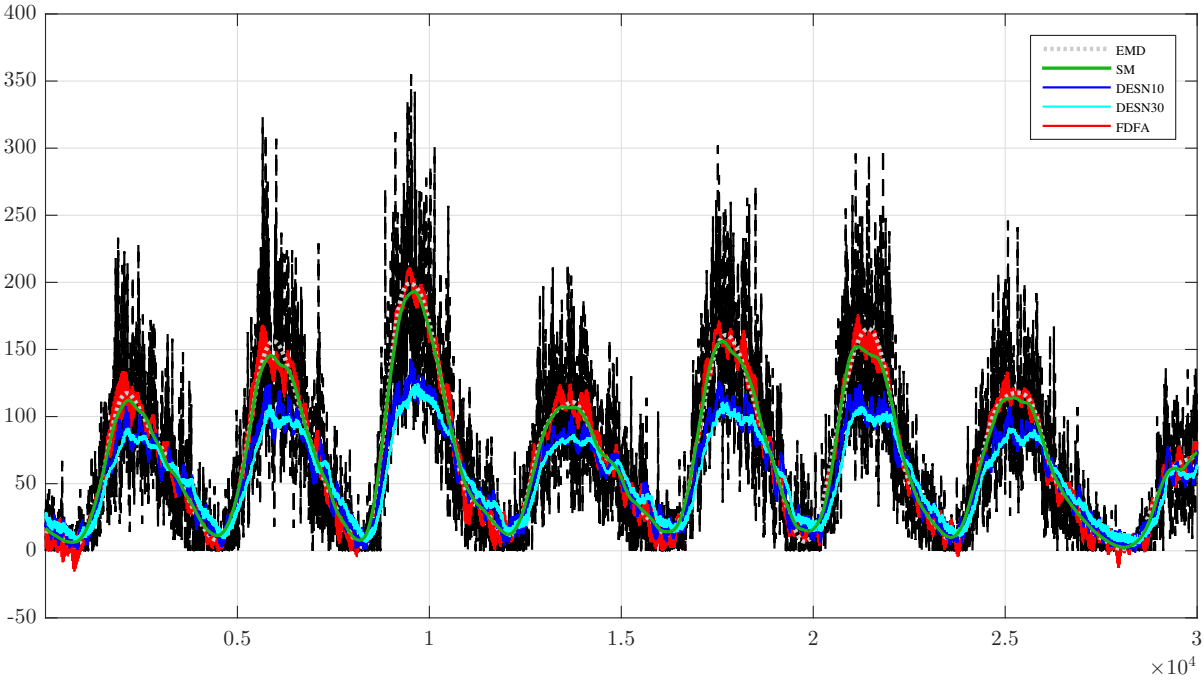


Figure 4.3. Colors online. Trends identified on the sunspot time series. The function depicted with black dashed lines represent the original time series. The colored lines represent the trends identified using EMD, FDFA, SM, and DESN with $k = 10$ and $k = 30$.

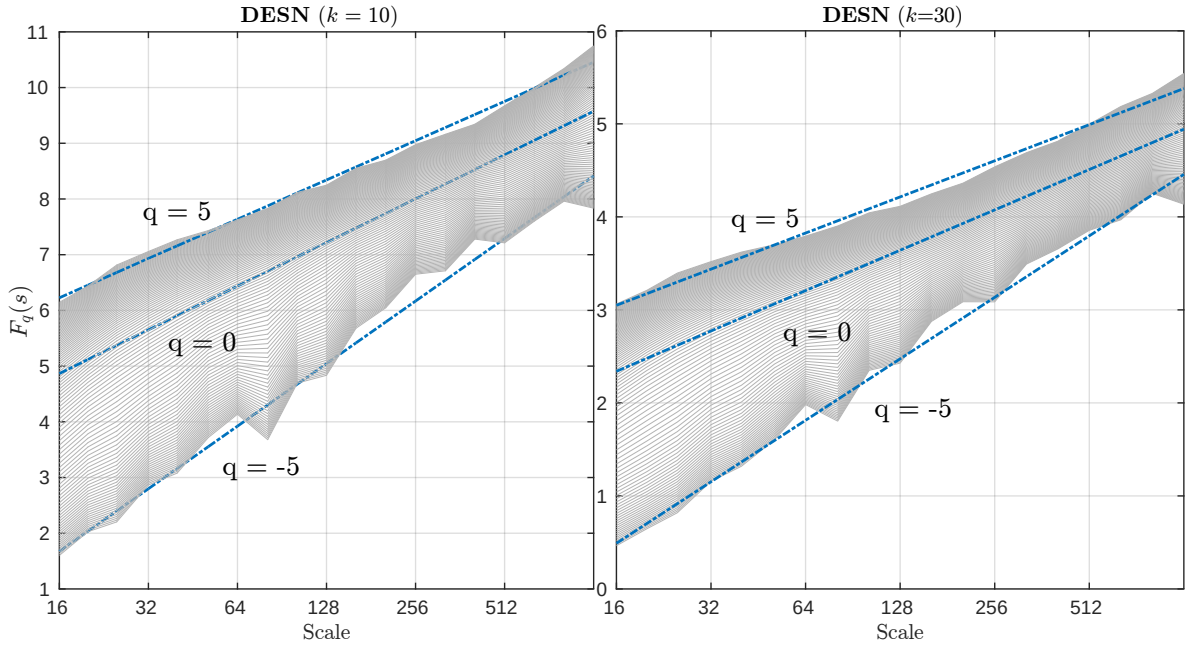


Figure 4.4. Scaling properties of the (detrended) sunspot time series obtained with DESN for two settings of the ensemble parameter k.

4.3 Discussion

In this Chapter, we have explored the possibility of identifying and removing trends in a given time series by means of echo state networks, a particular type of recurrent neural network. The proposed method, called DESN, allows to filter out trends with minimal assumptions and without performing a windowed fitting as proposed in other detrending approaches. This is possible by exploiting the capability of recurrent neural networks to learn and predict complex dynamical processes in order to separate the actual trend from its stochastic fluctuations. Our main assumption consists in considering the noise and trends components as processes with very different degrees of predictability. We exploited such an assumption as a separating criterion. Notably, we have used an ensemble of echo state networks as a filter, operating with a standard configuration and trained using linear regression for the readout layer. Many other approaches exist both for designing the reservoir and for training the readout [56, 150], which could be evaluated in future works depending on the specific problem at hand.

As a first benchmark, we have analyzed the performance of DESN and other detrending techniques taken from the literature on several synthetic time series generated using different types of trends and noise processes. The quality of the detrending has been evaluated by comparing the properties of the estimated noise with respect to the known ground truths. The evaluations of the Hurst exponents and the properties of the multifractal spectra on the detrended series have been performed with the multifractal detrended fluctuation analysis procedure, a consolidated method in the field of fractal analysis of time series. In most cases, the resulting fractal coefficients computed by DESN procedure agreed with the expected values and the noise self-similarity properties were preserved by the detrending operation. On the other hand, in several occasions other detrending methods were not able to perform a correct detrending, which resulted in an incorrect scaling of the fluctuation function. In general, DESN and a detrending method based on Fourier analysis have shown to be the most reliable methods in terms of detrending accuracy on the considered synthetic time series.

As a second test, we have analyzed the well-known sunspot time series, which is a multifractal time series that has been taken into account in several related works [54, 79]. Our experimental results suggest that the multifractal properties retrieved by using DESN were both qualitatively and quantitatively compatible with those suggested in other works taken from the literature. This further strengthens the validity of the proposed data-driven detrending method based on echo state networks.

Chapter 5

Generation of Protein Contact Networks

In Chapters 2 and 3 we investigated the peculiarities of the protein contact networks and assessed their differences with respect to the other networks and, in particular, the generative model PCN-S described in Sec. 1.6.1. We observed that there are still significant differences between PCN and PCN-S. Although many generative models have been developed in the still young network science discipline [28, 132], fewer and less established examples are available in the literature when focusing on formal representations of protein molecules [11, 31, 59, 121, 140, 141, 152]. In this Chapter we attempt at bridging this gap by designing a two-step generative model for PCNs. The first stage of our method starts from a modified version of the PCN-S model. The quest for a reliable and, most importantly, justifiable generative model for PCNs implies as a first step the identification of a target function. This allows for a unambiguous evaluation of the proposed model in terms of similarity of the simulated contact networks with the real PCNs. Inspired by the seminal works by Leitner [100], we considered here as target function to approximate the peculiar heat trace decay of PCNs shown in Sec. 1.4. Such a property is elaborated from the heat kernel [38, 95, 172], the graph-theoretical analogue of the well-known first-order differential equation describing diffusion of heat in a physical medium. We also evaluate the soundness of the proposed approach by focusing on mesoscopic analyses. In particular, we first study characteristics elaborated from the normalized Laplacian spectra of the generated networks. To complement this spectral analysis, we also analyze several topological descriptors of the resulting networks. Results show that the ensemble of networks generated with our method ends up into a significant improvement of similarity with real PCNs, as for both spectral and topological properties. However, a principal component analysis (PCA) of the considered topological descriptors reveals a gap with actual PCNs, specifically related to the shortest paths. The second step of the proposed method is hence designed to compensate this drawback. As a result, we show that we are able to achieve a further statistically significant improvement of the ensemble characteristics, without altering the global spectral properties of the first ensemble. As a byproduct of our study, we demonstrate that modularity, a well-known feature found in proteins as well as in many other biological networks, is not sufficient to explain the underlying network architecture of PCNs. This result is of particular interest, since it stresses the peculiar architecture of proteins that suitably merges conflicting features such as path efficiency and modularity.

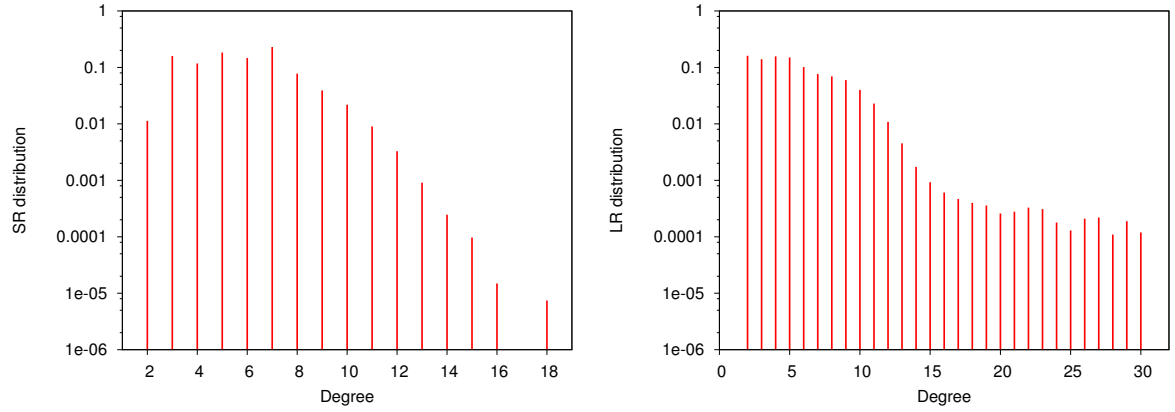
5.1 Dataset

In our study we start from three ensembles of networks. Each ensemble contains 100 networks of varying size (from 300 to 1000 vertices). The first two ensembles are the set of 100 PCN and 100 PCN-S presented in Sec 2.1. For the third ensemble we consider networks generated according to the recently-proposed scheme in Ref. Sah et al. [147]. Such a generation mechanism produces modular networks with controlled (i.e., used-defined) modularity and average degree values. In our study we consider these networks because, as we confirmed in Chapters 2 and 3, modular structures seem to be ubiquitous in biological networks. Indeed, modularity is considered to be at the basis of resilience and adaptability of biological networks. Accordingly, the third ensemble of Sah et al. networks is generated by copying modularity and average degree from the considered PCNs. By considering the Sah et al. ensemble we define a controlled frame or reference to assess the importance of modularity in PCN. We now proceed to describe the proposed generative method.

5.2 First step: the LMGRS generative model

Native contacts in folded proteins are in one way or another constrained by the covalent bonds due to the backbone. Therefore, a first interesting question that one would ask when designing a generative mechanism is "what is the effect of the backbone on the modular organization of a PCN?". To provide an answer to such question, we first define the notion of short range (SR) and long range (LR) contacts, that is, native contacts whose residues are, respectively, close and distant on the sequence (backbone). We chose 12 residues as threshold for SR contacts [137]. Fig. 5.1 shows the two separate degree distributions elaborated from the considered ensemble of varying-size PCNs. SR contacts denote a clearly different distribution with respect to those that are LR. Considering this fact and that PCNs do posses a modular architecture, one would be tempted to postulate a striking rule such as "SR contacts are intra-module while LR are inter-module links". If this rule was correct, it would be possible to design a generative mechanism accordingly, e.g., by connecting intra-module and inter-module links according to their specific (empirical) distributions. Nevertheless, such a possibility seems to be weakly supported by the following test. In Fig. 5.2 we show a graphical representations of two PCNs, denoted as "JW0058" and "JW0179". Those two networks contain roughly the same number of amino acid residues (around a thousand); JW0058 is made of two chains while JW0179 is derived from a single-chain polymer. To verify the above stated hypothesis, we need to consider a suitable criterion to find a partition of the vertices with maximum modularity. We again utilize the Louvain algorithm as defined in Ref. [27] to discover such partition. Results in Fig. 5.2 demonstrate that intra-module links (solid lines) are SR (drawn in red), in both cases, only around 55% of the times. This fact (that has been verified for a larger number of PCN) suggests to reconsider the possibility to follow such a SR/LR contacts characterization with respect to intra/inter module links. In addition, we found in our data that there is no trivial relation among the distance on sequence and the Euclidean distance among residues in the 3D space ($r \simeq 0.162$, full data not shown here). This complexity is expected as confirmed by the enormous research effort in predicting native contacts in proteins [12, 45, 57, 78, 89, 93, 114, 125, 126, 155].

Let us describe the proposed generative mechanism. Algorithm 1 conveys the pseudo-code of the procedure. The algorithm builds on the mechanism introduced by Bartoli et al. [20] and described in Sec. 1.6.1. Firstly, edges are deterministically added among any two residues at distance two on the sequence. In the model of Ref. Bartoli et al. [20], the second step consists in sequentially wiring pair of vertices with a probability that decays linearly with their distance on the backbone. In our model we instead substitute this linear probability with the observed



(a) SR contacts.

(b) Distribution of LR contacts.

Figure 5.1. Degree distribution of SR (5.1(a)) and LR (5.1(b)) contacts. Both distributions are provided in lin-log plots to improve visualization. SR contacts are determined by considering a distance on the sequence lower than or equal to 12 residues.

probability of the real PCN ensemble, evaluated from the empirical frequencies. In other words, the probability of wiring two vertices i and j that are placed at a distance $|i-j| = d$ on the backbone is equal to $n(d)/M_{\text{tot}}$, where $n(d)$ is the number of neighboring vertices at a backbone distance d in the PCN ensemble and M_{tot} the total number of edges. The empirical probability obtained from the PCN ensemble is shown in Fig. 5.3 (left) and an heat map of the log-probability of each edge on a typical adjacency matrix is shown in Fig. 5.3 (right). With this method we generate a set of networks where the number of nodes and number of edges match each network of the PCN ensemble. The generated graphs are referred to as LMGRS networks.

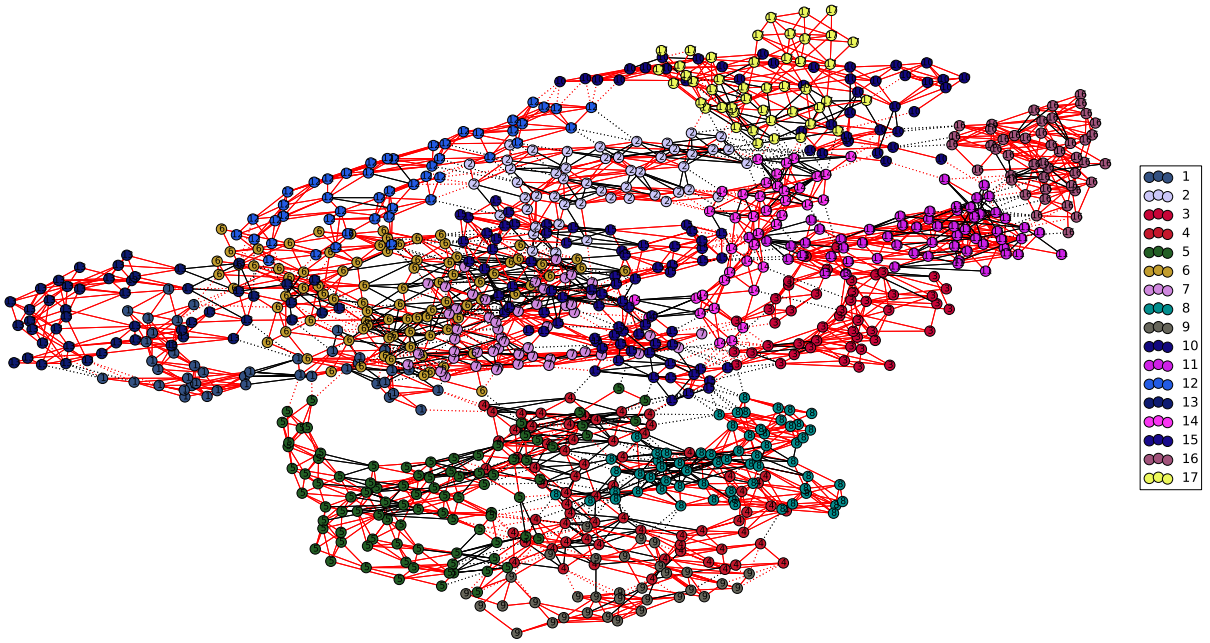
As shown in the following, this straightforward modification results in a considerable improvement under many aspects.

Algorithm 1 Pseudo-code of the proposed generative algorithm.

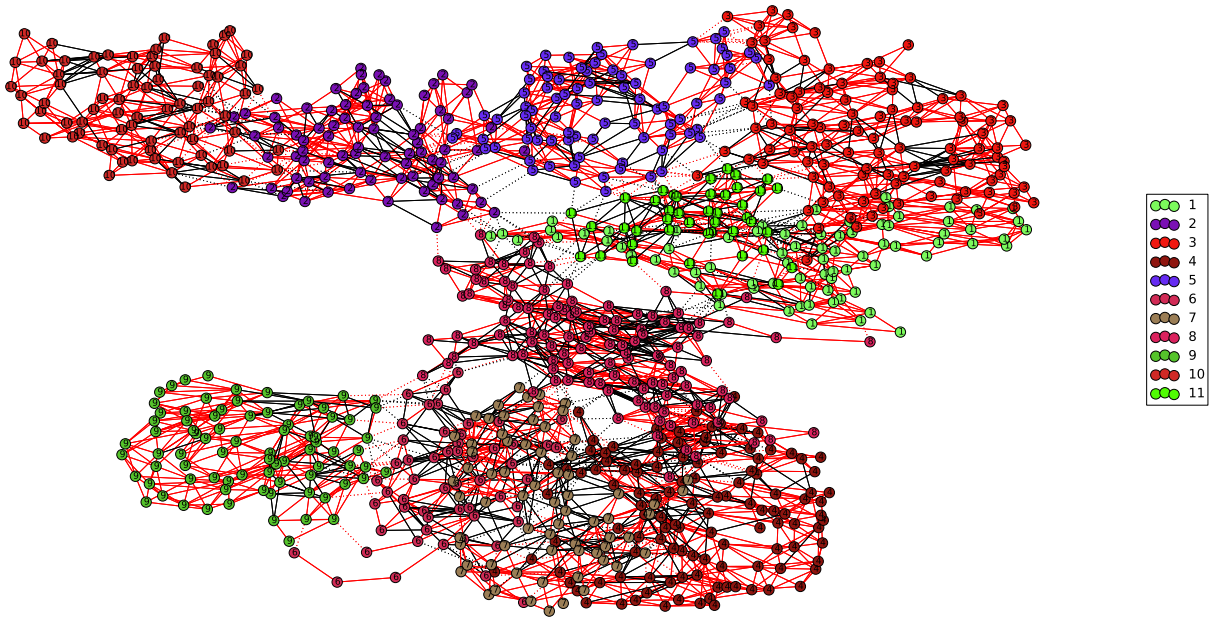
Require: Number of vertices, n, and edges, m
Ensure: A graph $G = (\mathcal{V}, \mathcal{E})$ with $n = |\mathcal{V}|$ and $m = |\mathcal{E}|$
1: Add n vertices in $\mathcal{V}$ with unique, progressive, numerical identifiers
2: Add backbone contacts in $\mathcal{E}$: connect all vertices v_i and v_j for which $|i-j| = 2$
3: Loop to add all remaining non-backbone contacts ⇝
4: **while** $|\mathcal{E}| < m$ **do**
5: Select two non-connected vertices v_i and v_j with probability $p(|i-j|)$ given by their distance $|i-j|$ according to the empirical distribution in Fig. 5.3
6: Add the undirected edge $e = (v_i, v_j)$ in $\mathcal{E}$
7: **end while**
8: **return** $G = (\mathcal{V}, \mathcal{E})$

5.3 Analysis of the LMGRS ensemble

We now analyze the spectral properties of the 3 ensembles of networks presented in Sec. 5.1 and compare them to the LMGRS ensemble generated with the method discussed in the previous Section. Fig. 5.4(a) shows the characteristic HT slopes decay (presented in Sec. 2.4) of the different ensembles. From the plot it is possible to note that LMGRS introduce a considerable improvement with respect to Bartoli et al. and Sah et al. ensembles; yet the PCN trend is not perfectly approximated. To obtain a more straightforward representation of the spectral



(a) JW0058 modules/links organization.



(b) JW0179 modules/links organization.

Figure 5.2. Classification of contacts by considering the SR/LR typology and the intra/inter module arrangement. The partition is derived by using the maximum modularity criterion. Vertex assignment to modules is represented using different colors; numerical module identifies are drawn in the legend and in the corresponding vertex labels. Solid lines denote intra-module links while dashed lines inter-module links. Black links denote LR contacts, while red links are SR. Please note that the length of the links in the figures respects the actual Euclidean distances of contacts. The assumption that LR contacts are mostly inter-module links (and accordingly, SR contacts are mostly intra-module links) seems to be disproved by those examples.

properties of these ensembles, we now proceed to analyze the spectral distributions of the networks of each ensemble. In particular, for each ensemble $\mathscr{C}$ we consider the flattened set $\boldsymbol{\lambda}_{\mathscr{C}} =$

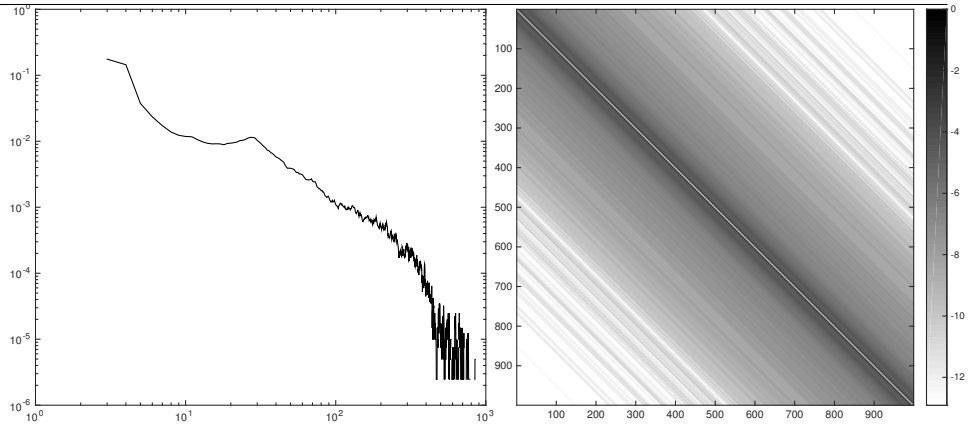
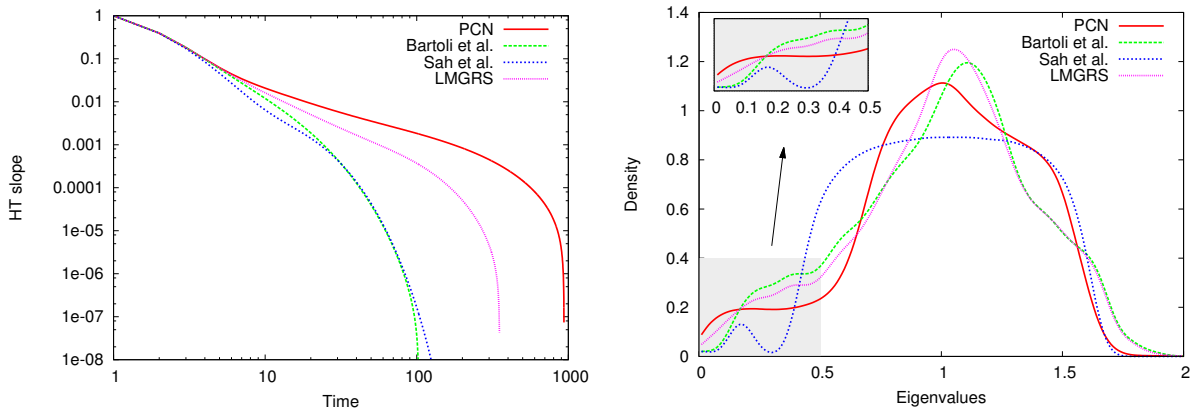


Figure 5.3. Distribution of backbone-distance probabilities for the contacts in the PCN ensembles (left) and relative heat map (right) with log-probabilities for each edge of the adjacency matrix.

$\{\lambda_1^{(1)}, \lambda_2^{(1)}, ..., \lambda_1^{(2)}, \lambda_2^{(2)}, ..., \lambda_N^{(100)}\}$ of all the normalized laplacian eigenvalues of all the networks of $\mathscr{C}$. We then estimate the global ensemble spectral distribution by performing a Gaussian Kernel Density Estimation on the set $\boldsymbol{\lambda}_{\mathscr{C}}$. A Kernel Density Estimation is a nonparametric way of estimating a probability density function of a random variable from a given set of observations [162]. In Fig. 5.4(b) are shown the ensemble spectral densitites of the 4 ensembles. The LMGRS ensemble density has clear similarities with the one of Bartoli et al., being the two based on the same algorithmic template. Nonetheless, by highlighting the left side of the distributions, it is possible to notice some improvement for a specific region (in-between 0 and 0.2). This region is particularly important for the modeling of PCN since it is well known that the smallest normalized eigenvalues are related to the modular organization of the networks and, in general, to the global network organization [9]. LMGRS ensemble offers a better approximation in the distribution of these eigenvalues, which explains the significant improvement observed for the HT decay. Now we move to the analysis of the ensembles by considering the representation of



(a) HT slopes decay.

(b) Spectral densities of ensembles.

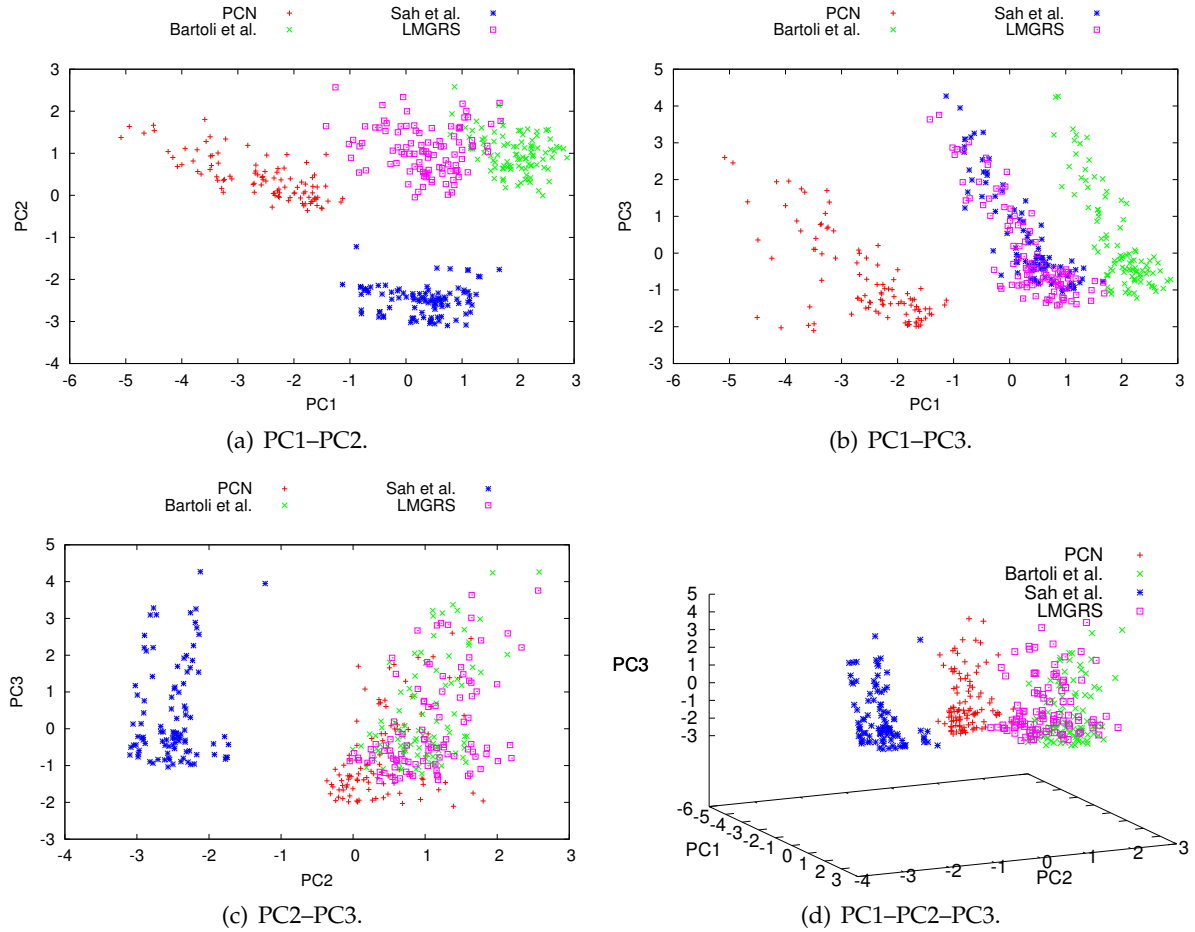
Figure 5.4. Ensemble HT slopes decay for the considered graphs 5.4(a) and related Laplacian spectral densities 5.4(b). The proposed LMGRS model clearly denotes more similar characteristics with respect to PCNs in terms of HT decay. Analogously, the LMGRS model induces a density of eigenvalues more similar to PCNs in the lower bands (see detailed plot), suggesting that the community structure is more suitably approximated. In comparison, the Sah et al. model instead does not mimic as well the spectral density of PCNs.

each graph as a numeric vector containing suitable features that characterize different aspects of the network topology, as done previously in Chapter 2. The topologic embedding vector generated for each network G is composed by the following components (see Sec. 2.2 for a complete description of their meaning):

- Modularity (MOD)
- Average Closeness Centrality (ACC)
- Average Shortest Path (ASP)
- Average Clustering Coefficient (ACL)
- Adjacency spectrum energy (EN)
- Laplacian spectrum energy (LEN)
- Random Walk Entropy (H)
- Graph Ambiguity (A)

To offer a synthetic visualization, in Fig. 5.5 we perform a Principal Component Analysis of the resulting vector space and show the first three components. The first three PCs explain $\simeq 92\%$ of the entire data variance (PC1 $\simeq 39\%$, PC2 $\simeq 30\%$, and PC3 $\simeq 23\%$) and therefore they are offer a complete description of the space; the component loadings (Pearson correlation coefficients between original descriptors and components) are reported in Tab. 5.1. The loadings on the PC offer an easily interpretable scenario, where PC1 is mostly explained by the path distribution (ACC and ASP) and the local clustering (ACL). As expected ACC negatively scales with both ASP and ACL, so pointing to the fact that ACL decreases the efficiency of signal transmission across the network (positive correlation with ASP). Thus, high values of PC1 corresponds to architectures with high characteristic length (slow information transmission), while low values of PC1 point to graphs with high closeness centrality (ACC) and thus relatively efficient signal transmission. PC2 is mainly correlated with MOD, A and H, with A going in the opposite direction with respect to the other two descriptors. This corresponds to the fact the regularity of a graph decreases as the modularity increases; it is also well-known that modularity affects random walks behavior, explaining the positive correlation with H. PC3 is entirely described by the spectra of the adjacency and Laplacian matrices (respectively indicated by EN and LEN).

The PCs are linearly independent by construction, so the above results clearly indicate that the dataset can be described by three autonomous topological features: 1) path length and local clustering (PC1); mesoscopic modularity (PC2); and 3) spectral properties (PC3). The particular mixing of these independent features varies across the different ensembles. Let us now focus on the PCA subspace spanned by PC1-PC2 (Fig. 5.5(a)). It is possible to note that the LMGRS ensemble introduces an improvement in PC1, which as explained before, encodes contributions in terms of path distribution and local clustering. A very interesting scenario can be observed when considering the projection given by PC1-PC3 (Fig. 5.5(b)). In fact, when PC2 is not considered Sah et al. and LMGRS networks become very similar to each other, and entirely different from the ensemble of Bartoli et al. To summarize, it is worth pointing out that the average Euclidean distance among the LMGRS and PCNs networks represented in the three-dimensional PCA space is significantly inferior ($p < 0.0001$) with respect to the distances among Bartoli et al. and PCNs (3.13 vs 4.69 with standard deviations 0.75 and 0.68, respectively).



(a) PC1–PC2. (b) PC1–PC3.

(c) PC2–PC3. (d) PC1–PC2–PC3.

Figure 5.5. PCA of the topological descriptors calculated on the four ensembles of protein graphs. LMGRS model is an improvement with respect to Bartoli et al. [20] also considering classical TD.

Table 5.1. Principal component loadings.

	PC1	PC2	PC3
MOD	0.26148	**0.78872**	0.13409
ACC	**-0.97835**	-0.15695	-0.11588
ASP	**0.92838**	0.26494	0.09270
ACL	**0.89098**	-0.22533	-0.12227
EN	0.01862	0.27707	**0.95810**
LEN	0.04354	-0.27981	**0.94213**
H	0.05171	**0.99519**	-0.04393
A	0.09073	**-0.84835**	0.06463

5.4 Second step: the LMGRS-REC reconfiguration procedure

From PCA space snapshots we deduce that the LMGRS ensemble is a considerable improvement with respect to the others, even if there is still a gap to be filled with respect to PCN. In particular, LMGRS networks present a too small value of the average shortest path length, i.e. the small-world signature is too strong. This fact explains the differences observed in path

distribution and modularity, since the path efficiency and modular properties are two conflicting features in networks, that is, modular organization is progressively lost as the network becomes more and more interconnected and the average shortest path decreases. To this end we now consider another ensemble derived by post-processing LMGRS networks with the following edge reconfiguration process. Given a graph $G = (\mathscr{V}, \mathscr{E})$, the reconfiguration step is primarily meant to lower the small-world signature in G. This is done by iteratively rewiring edges in $\mathscr{E}$ according to their edge-betweenness value. The pseudo-code of the edge reconfiguration algorithm in shown in Algorithm 2. The reconfiguration process rewires edges with high edge betweenness centrality[1], since those edges have a direct impact on path efficiency, and accordingly also on the modular organization. At each iteration, the edge with maximum edge-betweenness is removed and it is re-attached to two randomly chosen vertices at a backbone distance given by the empirical distribution shown in Fig. 5.3. This process is repeated until a suitable convergence criteria is met. In our case, we considered a number of iterations (50) that resulted in a statistically significant improvement of the ensemble features that we observe. A reconfigured graph $\hat{G}$ is obtained at the end of the reconfiguration loop. Notice that we insure connectedness for all $\hat{G}$. The loss of small-world signature in $\hat{G}$ is primarily verified with the ASP increase (see Fig. 5.6(b) for an example), which is a consequence of the targeted rewiring of edges with maximum edge-betweenness. We apply this reconfiguration step to all the generated LMGRS networks, obtaining the so-called LMGRS-REC ensemble. In Fig. 5.6(a) we show the

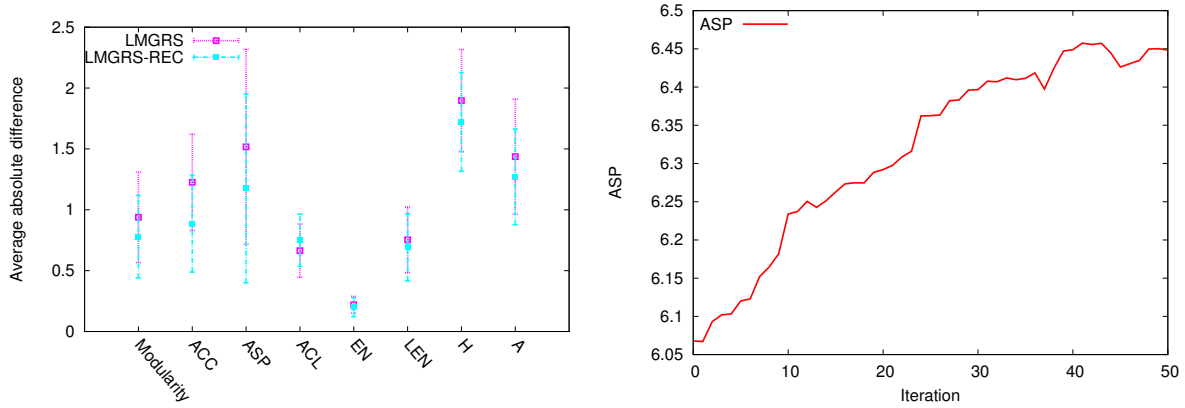
Algorithm 2 Pseudo-code of the proposed edge reconfiguration algorithm.

Require: A graph $G = (\mathscr{V}, \mathscr{E})$ with $n = |\mathscr{V}|$ and $m = |\mathscr{E}|$
Ensure: A modified graph $\hat{G} = (\mathscr{V}, \mathscr{E})$ with $n = |\mathscr{V}|$ and $m = |\mathscr{E}|$
1: **loop**
2: Calculate the edge-betweenness measure for all edges in $\mathscr{E}$
3: Let $e_{\max}$ be the edge with maximum edge-betweenness. Remove $e_{\max}$ from $\mathscr{E}$
4: Select two non-connected vertices v_i and v_j with probability $p(|i-j|)$ given by their distance $|i-j|$ according to the empirical distribution in Fig. 5.3
5: Add the undirected edge $e = (v_i, v_j)$ in $\mathscr{E}$
6: **end loop** when stop criterion is met
7: **return** $\hat{G} = (\mathscr{V}, \mathscr{E})$

detailed changes of several topological properties for the LMGRS and LMGRS-REC with respect to the PCNs. The figure reports, for each descriptor, the average absolute difference calculated for each graph of the respective ensembles with respect to the PCN graphs; standard deviations are reported as vertical bars. Results show that the reconfiguration algorithm performs well as for statistical significance of differences with respect to the LMGRS ensemble, assessed via t-test with the usual 5% threshold. In particular, as desired reconfigured networks denote more similar ASP and ACC. As expected, such improvements for the shortest paths have a direct influence on the global modularity. In fact, MOD is significantly improved. This is a direct consequence of the fact that path efficiency and modularity are features to be considered in a trade-off. ACL similarity improves as well, denoting a better approximation of the local cluster structure of PCNs. It is important to note that differences for EN and LEN are not statistically significant. This fact tells us that spectral properties of the reconfigured networks are not significantly altered. Fig. 5.7(b) offers a visual confirmation of this fact. In fact, the spectral densities for LMGRS and LMGRS-REC reported in the figure are almost identical. However, it is worth discussing the HT slopes shown in Fig. 5.7(a). From the figure, it is possible to notice a slight divergence among LMGRS and LMGRS-REC for large-time instants. This is due to the difference in magnitude of the first non-zero eigenvalue of the normalized Laplacian, which particularly

[1] the edge betweeness centrality is the betweeness centrality associated to an edge instead of a node. See Sec. 1.2.1 for a definition of the betweeness centrality

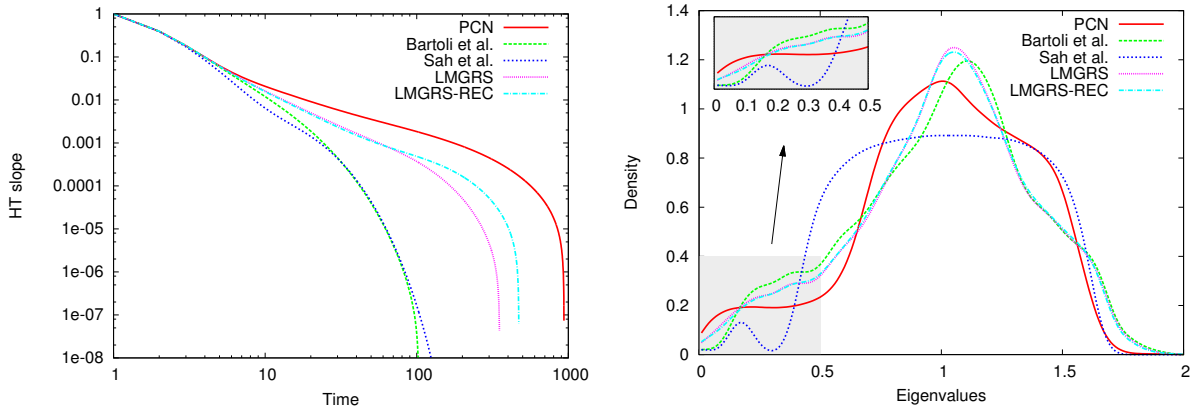
influences the asymptotic HT behavior. Such a difference is a byproduct of the designed edge reconfiguration algorithm, which focuses on rewiring edges with high edge-betweenness: those are most likely connections among different densely connected communities.



(a) Average absolute differences.

(b) ASP over iterations.

Figure 5.6. Average differences for each topological descriptor with respect to PCN (5.6(a)) and their standard deviations. We are able to modify, among the other factors, the small-world character of the generated networks without significantly affecting the spectra of the adjacency (EN) and Laplacian (LEN) matrices. Statistical significance of differences is assessed via t-test. Fig. 5.6(b) shows the ASP of a sample graph during the reconfiguration process.



(a) HT slopes.

(b) Spectral densities.

Figure 5.7. Same as Fig 5.4 but including also reconfigured LMGRS.

5.5 Discussion

In this Chapter, we proposed a two-step generative model for protein contact networks. For the first step, we partially took inspiration from the work of Bartoli et al. [20], whose idea is to generate contact graphs by first adding backbone contacts deterministically (considering adjacent residues along the sequence). Successively, a number of additional contacts are added with a probability inversely proportional to the residue distance along the sequence. Here we modified this part by considering the actual empirical probability distribution of contacts with respect to the sequence distance, derived from an ensemble of E. coli proteins.

We analyzed our generative method by considering three additional ensembles composed of 100 varying-size protein contact networks. We focused on a mesoscopic analysis, that is, we primarily investigated the soundness of the models by considering information derived from the eigendecomposition of the normalized Laplacian. Results showed that the proposed method approximates with better precision the behavior of actual protein contact networks in terms of characteristic diffusion time. We considered also several common topological descriptors. This last analysis pointed out that our method, as well as the others, does not approximate sufficiently well the path distribution. To this end, we designed an edge reconfiguration algorithm to be used as the second step of the proposed generative method. We then generated an additional ensemble of reconfigured networks, which showed statistically significant improvements with respect to the initial one.

We considered an ensemble composed of graphs synthesized according to a recently-proposed mechanism [147], designed to construct a network with specified modularity and degree profiles. Notably, we reproduced the modularity and degree values from the actual protein contact networks herein considered. Results demonstrate that modularity, when hard-coded into the networks, does not explain alone the actual architecture of proteins. In fact, we modularity should be considered as an emergent property of such networks, which is suitably optimized in a trade-off with the conflicting feature of path efficiency. In our model, an increased modularity emerged from the peculiar PCNs mesoscopic wiring obtained from their empirical contact distribution at increasing distance length: a simple linear decrease in contact frequency at increasing sequence distance does not allow to reach the typical modularity of real proteins. The fine tuning of long-range contacts allows for directly intervening on both modularity and path efficiency balance, so confirming the crucial importance of long-range contacts in folding process [37, 167]. A sound generative mechanism for protein contact networks is of utmost importance in current researches in protein science. The possibility to learn in a data-driven fashion an effective model for protein contact networks would allow to easily generalize other instances of such networks. This perspective could be interesting also for protein engineering purposes [42]. Morevoer, the theoretical study of networks promises to pave the way for the discovery of universal principles at the basis of protein organization.

Chapter 6

Optimization of the LMGRS networks

In the previous Chapter, we attempted at reproducing the peculiar spectrum of PCNs by an incremental approach: synthetic networks are generated by the LMGRS generative model based on known features of PCN, and the distance of the generated networks from real PCNs spectra was estimated. Successively, the LMGRS are reconfigured in order to improve their similarity in average shortest path with the real PCN, obtaining the LMGRS-REC networks.

In this Chapter, we follow an alternative approach to the reconfiguration step. We focus here on the spectral similarity [69] between the generated matrices and the experimental PCNs under consideration. To this end, we exploit a evolutionary optimization scheme in which networks are iteratively selected and modified according to a suitable fitness function. A similar problem has been discussed in the works of Refs. [39, 84]. In these works, the problem consists in reconstructing a matrix with a given normalized Laplacian spectrum. Whether and in which cases the spectrum univocally determines the corresponding graph is still an open and difficult problem, which is known as *inverse spectral problem* [77, 164, 165]. In fact, a straightforward procedure for reconstructing a graph adjacency matrix from a given Laplacian spectrum is still unknown. Moreover, there are known classes of co-spectral graphs, i.e., non-isomorphic graphs exhibiting identical Laplacian spectra, for which such an operation would be impossible to perform. Nevertheless, such classes are believed to be sufficiently rare in the networks space and thus the Laplacian spectrum can be considered as a meaningful representation of most topological properties of a graph and hence of the modeled system. However, in this work we are interested in recognizing the common principles behind *all* the PCN topologies, so our task is to find a graph that *statistically* resembles a typical, realistic PCN, without referring to any particular protein. Therefore our objective is not to reconstruct a network with a given normalized laplacian spectrum, but instead we aim at generating a network with a given spectral density. In this context, we recall the concept of *spectral classes* of networks, mentioned in Sec. 2.4 (see Refs. [14, 15, 70]), i.e., groups of networks which share similar spectral densities. The proposed PCN generation method optimizes candidate solutions (i.e., graphs) in order to minimize the spectral distance with respect to the PCN class. We show that our method is capable of producing networks of varying, user-defined size, all having the very same spectral density characteristic of the considered PCN ensemble. Finally, we complement the analysis of the generated network spectra by studying the correlation properties of their Laplacians. This is performed by borrowing the tools developed in the field of Random Matrix Theory [118], which have been recently applied to network science [87, 88, 119].

6.1 Spectral classes

The boundedness of the spectrum of the normalized graph laplacian operator in $[0,2]$, which is independent of the number of vertices N, makes it a suitable tool to compare networks of different sizes. Indeed, it is possible to treat a graph Laplacian as a random matrix [88] with a well-defined spectral density $\rho(\lambda)$. As discussed by Gu et al. [70], and mentioned in Sec. 2.4 matrices sharing a common normalized Laplacian spectrum density $\rho(\lambda)$ form a *spectral class*. To estimate the spectral density of a given matrix, we employ the well-known kernel density estimation (KDE) method equipped with a Gaussian kernel. Starting from the spectrum $\lambda(\mathscr{L}) = \{\lambda_i\}$, we obtain the continuous function

$$\rho(x) = \frac{1}{N}\sum_{i=0}^{N-1} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\lambda_i)^2}{2\sigma^2}\right), \tag{6.1}$$

where σ is the kernel bandwidth. The spectral density $\bar{\rho}_{\mathscr{C}}(x)$ of a spectral class $\mathscr{C} = \{G_1, G_2, ...\}$, or *spectral class density* (SCD), is defined as

$$\bar{\rho}_{\mathscr{C}}(x) \equiv \langle \rho_G(x) \rangle_{G\in\mathscr{C}}, \tag{6.2}$$

where ρ_G is the spectral density of a graph $G \in \mathscr{C}$. Notice that the exact spectrum of a spectral class is not well-defined. The main assumption of this work is that the SCD of a set of networks preserves (part of) the common characteristics in the structural organization of all networks of the set. We also define the spectral distance $d(G_1, G_2)$ between two graphs G_1 and G_2 as the ℓ^2 distance between their spectral distributions:

$$d(G_1, G_2) := \int_0^2 [\rho_1(x) - \rho_2(x)]^2 \, dx, \tag{6.3}$$

where $\rho_1(x)$ and $\rho_2(x)$ are, respectively, the estimated densities of $\lambda(G_1)$ and $\lambda(G_2)$. Given the boundedness of the normalized Laplacian spectrum, this transformation to a continuous distribution allows us to compare spectra with a different number of eigenvalues in a natural way, hence overcoming the problem of comparing networks having different sizes. Analogously, the distance in eq. (6.3) can be readily extended to the distance of graph from a spectral class, by writing

$$d(G_1, \mathscr{C}) := \int_0^2 [\rho_1(x) - \bar{\rho}_{\mathscr{C}}(x)]^2 \, dx. \tag{6.4}$$

with $\bar{\rho}_{\mathscr{C}}(x)$ denoting the SCD of class $\mathscr{C}$. The notion of spectral class is closely related to the concept of random matrix ensembles, which will be formalized in the next section.

6.1.1 Random matrix theory

Random Matrix Theory (RMT) studies the properties of ensembles of random matrices [118]. RMT originated in nuclear physics for the analysis of the energy spectra of atomic systems. According to RMT, the matrices of an ensemble can be seen as independent realizations of a generalized random variable that possess common spectral and structural properties. From a probabilistic analysis of the spectrum of such matrices it is possible to extract important universal properties of the ensemble and of the underlying generating process. One of the most famous results on the universality in RMT is the Wigner's semicircle law, which demonstrates that the spectral distribution of a large random matrix converges to a semicircular form over the real line. The most common universal ensembles studied in RMT are the Gaussian Orthogonal

Ensemble (GOE), Gaussian Unitary Ensemble (GUE), and Gaussian Symplectic Ensemble (GSE), each one corresponding to different universality classes [118]. RMT has been successfully applied also to matrices representing complex networks [88]. One of the most commonly used statistics in RMT is the nearest neighbors spacing distribution (NNSD), which measures the degree of repulsion between neighbouring eigenvalues in the spectrum. Computing the NNSD of a spectrum allows to derive a universal law that measures the degree of repulsion of the eigenvalues, regardless of their distribution over the real line. In fact, areas of higher (lower) spectral density would lead to erroneously consider eigenvalues as more (less) attractive, and hence hide the universal fluctuation properties (i.e., those independent from the particular spectral distribution) of eigenvalue spacings.

For this reason, given a spectrum $\{\lambda_i\}$ with spectral distribution $\rho(x)$, it is common to consider a new spectrum given by the values assumed by the cumulative distribution of $\rho(x)$ evaluated at the points λ_i,

$$\bar{\lambda}_i = F(\lambda_i) = \int_0^{\lambda_i} \rho(x)dx. \tag{6.5}$$

This operation is called *unfolding* [118] and it allows to obtain a uniformly-distributed spectrum with unitary mean spacing between eigenvalues. The spacings between eigenvalues are then calculated as $s_i = \bar{\lambda}_{i+1} - \bar{\lambda}_i$. Since the functional form of the cumulative distribution is not known when dealing with empirical data, it is customary to approximate $F(\lambda_i)$ by numerically fitting a polynomial curve to the empirical eigenvalue distribution [4]. This operation is similar to the detrending procedures employed in time series analysis for analyzing fluctuations of nonstationary processes [63]. In the study of atomic energy spectra, Wigner hypothesized that the closer one gets to a level, the smaller the probability becomes of finding another one. This repulsion is well-described by the probability distribution [87]

$$P_\beta(s) = As^\beta \exp\left(-Bs^{\beta+1}\right), \tag{6.6}$$

which is called Wigner's surmise for $\beta = 1$ or Brody formula in its generalized form. In (6.6) $A = (1+\beta)B$ and $B = \left[\Gamma\left(\frac{\beta+2}{\beta+1}\right)\right]^{\beta+1}$ are two parameters; $\Gamma(\cdot)$ is the Gamma function. $P_\beta(s)$ is the probability of finding a spacing s between two consecutive eigenvalues $\bar{\lambda}_i$ and $\bar{\lambda}_{i+1}$. The values $\beta = 1, 2, 4$ correspond to the GOE, GUE, and GSE ensembles, respectively. When $\beta \to 0$, instead, Eq. (6.6) approaches the Poisson distribution, implying that the eigenvalues in the spectrum are completely independent of one another.

To the best of our knowledge, all ensembles commonly studied in RMT consist of matrices of the same size [63]. In this work, we will relax this constraint by studying ensemble properties of several spectral classes of normalized Laplacian matrices of graphs having different dimensions. Intuitively, this is allowed by the fact that the NNSD depends only on local fluctuation properties of the eigenvalues, i.e., their first neighbours and not on the global organization of the spectrum.

6.2 Datasets

We consider three sets of networks, namely PCN, Bartoli et al. networks and LMGRS networks. These sets are the same ensembles of networks considered in Chapter 5. More specifically, the dimension and connectivity of each graph in each set has been chosen to resemble the characteristic of a corresponding protein in the PCN set, for a total of 100 graphs for each set. For each set of networks, we computed their SCD with the approach discussed in Sec. 6.1. We refer to these densities as $\bar{\rho}_{\text{PCN}}(x)$, $\bar{\rho}_{\text{Bartoli}}(x)$, and $\bar{\rho}_{\text{LMGRS}}(x)$, respectively.

6.3 PCN reconfiguration by means of genetic algorithms

The objective of the reconfiguration is to optimize the graphs obtained through the LMGRS method explained in Sec. 5.2. In particular, we aim for the minimization of the dissimilarity between the normalized Laplacian spectrum of the graph to be optimized and the SCD of the real PCNs. Our method is able to optimize one network at a time, having a specific user-defined number of vertices N and edges E. A symmetric graph is composed of at most $E = N(N-1)/2$ edges, where N is the number of vertices of the graph. The backbone consists only in the contacts between the amino acids at distance 2, i.e., edges $(i,\ i+2)$, as the first neighbors are trivial contacts given by the linear structure which are not informative of the global contact organization of the protein and hence unnecessary for our representation. Since the backbone contacts are fixed a priori, the remaining degrees of freedom for the contact maps are $\tilde{E} = N(N-1)/2 - 2N + 2$, i.e. the total number of unconstrained possible edges, so the search space dimension is $2^{\tilde{E}}$. For the networks under consideration, with a number of vertices varying from 300 to 1000, the search translates to a combinatorial optimization problem with a considerably large search space.

The objective function to be minimized must be representative of the distance between the Laplacian spectrum $\lambda(G)$ of a candidate solution graph G and the SCD $\bar{\rho}_{\mathrm{PCN}}(x)$ of PCN. A convenient form for the objective function is thus

$$\tilde{d}_w(G, \mathtt{PCN}) = \int_0^2 w(x) \left[\rho_G(x) - \bar{\rho}_{\mathtt{PCN}}(x)\right]^2 dx. \tag{6.7}$$

where $\rho_G(x)$ is the estimated spectral density of the normalized laplacian spectrum of graph G, as explained in Sec. 1.4. The distance $\tilde{d}_w(\cdot,\cdot)$ reduces to $d(\cdot,\cdot)$ of eq. (6.4) when $w(x) = 1$, as defined in Eq. (6.3). The motivation for the choice of such a distance function will be given in Sec. 6.4. To follow the common convention in evolutionary algorithms design, we define the fitness as a function to be maximised, so for a given graph G its fitness function $f(G)$ corresponds to

$$f(G) = -\tilde{d}_w(G, \mathtt{PCN}). \tag{6.8}$$

Since the evaluation of the spectrum of a generic matrix is not available in closed form, and the objective function (6.8) depends on the eigenvalues of the normalized laplacian matrix, the optimization has to be carried out with a derivative-free optimization method. For this reason, we employ a genetic algorithm [67] equipped with custom operators in order to perform the search in the matrix space. Every candidate solution is an adjacency matrix that is represented with a binary genetic code, where each gene encodes the presence of a specific edge of the contact map (we remind that only non-backbone contacts are optimized). The initial population of the genetic algorithm is composed by the set of LMGRS adjacency matrices of equal size N and connectivity E. The flow of the genetic algorithm, as implemented in this work, is reported below:

1. A population of N_{pop} individuals (adjacency matrices) is generated according to the LMGRS method presented in Sec. 5.2. The binary genetic code and corresponding fitness of each individual is evaluated.

2. The population is grouped in $N_{\mathrm{pop}}/2$ randomly chosen pairs of individuals. For each pair, with probability μ_{cross}, a crossover operator is applied and two new individuals - the *offspring* - are generated, for an average total of $N_{\mathrm{pop}} * \mu_{\mathrm{cross}}$ new individuals. The task of the crossover is to attempt to generate individuals that share advantageous traits of both parents.

3. For each individual of the newly generated offspring, a mutation operator is applied with probability μ_{mut}. This operator is needed to avoid trapping the evolution in local minima and gradually refresh the gene pool - i.e. the unconstrained edges configurations.

4. The fitness value, related to the objective function to be minimized, is calculated for each individual of the actual population composed by the previous population and for the newly-generated offspring, for an average total of $N_{\text{pop}} + N_{\text{pop}} * \mu_{\text{cross}}$ individuals. The population is subsequently ordered by decreasing fitness values.

5. The new population is formed by keeping only the N_{pop} fittest individuals, i.e. those with higher fitness values, among the previous population and the offspring, while the remaining individuals are removed.

6. Steps 2 to 5 are iterated until convergence of the fitness value, evaluated as a variation threshold over a fixed number of iterations, or until a maximum number of iterations has been reached.

In the following, the objective function and the main operators of the genetic algorithm are described more in detail.

6.3.1 Genetic algorithm operators

In the preliminary stage of this study, several different operators of mutation and crossover have been tested for the genetic algorithm. However, for the sake of brevity, in the following we report the details only about the operators that have been actually used in this work.

Mutation For the mutation we used a custom version of the random shuffle operator. As described above, each individual is randomly selected for the mutation with probability μ_{mut}. When an individual is selected, each unconstrained edge of the corresponding graph (i.e., genes with value 1) has a probability of being randomly relocated with probability μ_{p}. The new position of the edge in the adjacency matrix is then extracted according to the empirical distribution of backbone distances already used for the creation of the LMGRS graphs and shown in Fig. 6.1.

Crossover The crossover operator has been designed in a way to preserve secondary structure elements, namely α-helices and β-sheets. In fact, as explained in Sec. 1.6, adjacency matrices of proteins are typically characterized by a composition of similar higher-order structures, which should be preserved in a crossover operation. However, the preliminary detection process of known higher-order structures would add a considerable overhead to the crossover operation, due to the need of calling each time an inexact motif recognition procedure, and would exclude any other kind of network motifs potentially useful for the organization of the protein structure but unknown *a priori*. For this reason, we introduce a more general heuristic for the crossover, which we refer to as *chessboard crossover*. To preserve the higher-order organization of the mating individuals, we divide each parent's matrix in squares of equal size, as shown in Fig.6.1, where the square's side length is selected with uniform probability in the range $[l_{\text{min}}, l_{\text{max}}]$, where l_{min}, l_{max} are two user-defined parameters. Each diagonal square on the matrix represents a randomly-defined backbone community, i.e., a subgraph with edges connecting only internal nodes (intra-community links). Off-diagonal squares, instead, encode edges between different backbone communities. This subdivision is justified by the fact that α-helices are secondary structures that corresponds to edges between close amino acids on the backbone, as shown

in Fig. 1.2, and hence correspond mainly to backbone communities. On the other hand, β-sheets are composed by several parallel or anti-parallel sections of the backbone connected laterally (β-strands), and can be composed by inter-community as well as intra-community links. It is important to point out that the communities defined here are not intended in the common graph theoretical sense as defined, for instance, in [176]. The two offspring are then generated by randomly selecting the squares from each parent. In particular, for each square a random variable distributed uniformly between 0 and 1 is extracted. If the extracted value is $< 0.5 (\geqslant 0.5)$ then the first (second) child inherits the corresponding genes from the first parent and the second (first) child inherits the genes from the second parent.

While the mutation operator preserves the number of edges of the matrix, the offspring resulting from a crossover have in general different connectivity. However, since this operation is symmetrical and we start from graphs with equal connectivity E, we expect the mean connectivity to remain approximately the same. This assumption has also been confirmed by empirical evaluation. The individuals are selected in pairs for the crossover with probability μ_{cross}.

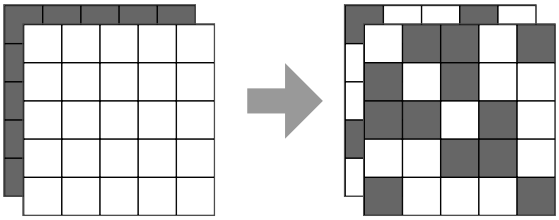


Figure 6.1. Chessboard crossover between adjacency matrices. Each parent is divided in squares of size c representing intra-community links (diagonal squares) and inter-community links (off-diagonal squares).

6.4 Results

The three sets of networks considered – PCN, Bartoli and LMGRS – are each composed of networks which exhibit very similar spectral distributions, as shown in Fig. 6.2. To quantify this 'spectral compactness' of the ensembles and to compare their relative spectral differences we computed the spectral distance between the normalized Laplacian of every graph and the SCD of every set among PCN, Bartoli, and LMGRS. By considering each pair of classes $\mathscr{C}_u, \mathscr{C}_v \in \{\text{PCN, Bartoli, LMGRS}\}$, we calculated a matrix of average distance values given by:

$$d_{u,v} := \left\langle d\left(\rho_L(x), \bar{\rho}_{\mathscr{C}_v}(x)\right)\right\rangle_{L\in\mathscr{C}_u}. \tag{6.9}$$

Here $d_{u,v}$ is the average over all matrices $\tilde{L} \in \mathscr{C}_u$ of the spectral distance between $\tilde{L}$ and the SCD $\bar{\rho}_{\mathscr{C}_v}(x)$ of the class $\mathscr{C}_v$. The results are shown in Table 6.1. The elements on the diagonal can be interpreted as the compactness of each set from a spectral point of view, while off-diagonal terms describe the spectral overlap between different sets. It is worth noting that the definition in Eq. (6.9) implies that the matrix $d_{u,v}$ is not necessarily symmetric.

PCN is the most separated and compact cluster, while Bartoli and LMGRS denote some overlap. This is expected, since synthetic ensembles have been generated with fairly similar criteria. On the other hand, the compactness of the PCN ensemble is a hint of the basic tenet of our study, i.e. that a "typical" and largely invariant topological model of a proper protein molecule does exist in the networks configuration space.

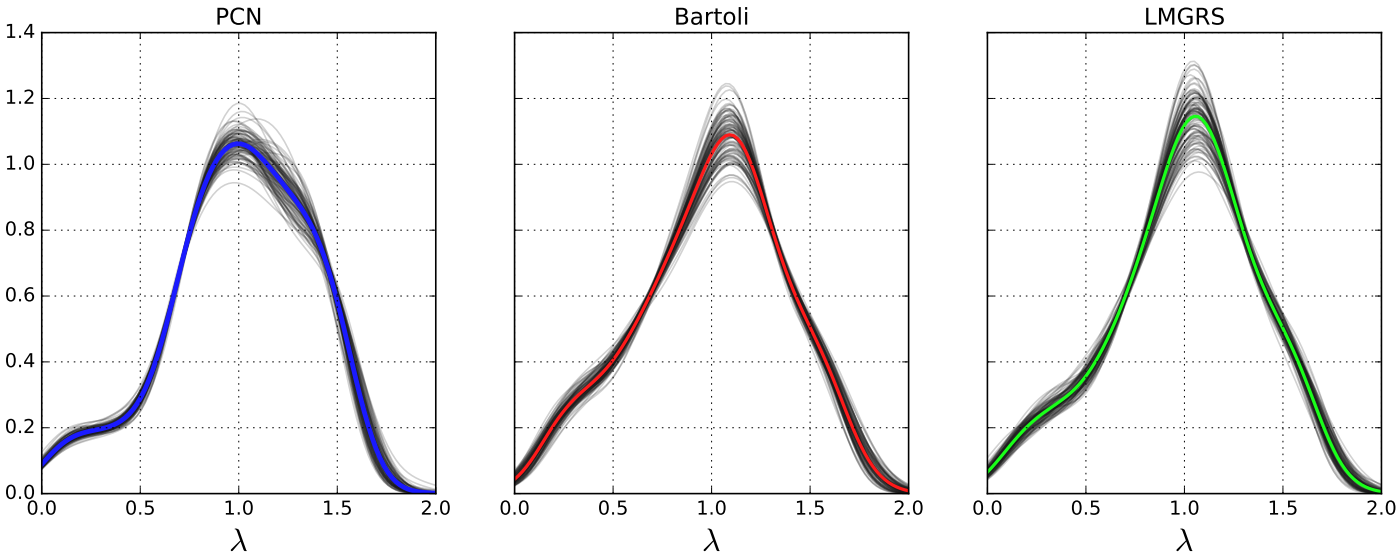


Figure 6.2. Colors online. Superposition of all the estimated spectral densities of the PCN, Bartoli and LMGRS ensembles. The colored curves represent their respective SCDs. Notice that for each ensemble all the curves follow a similar pattern.

To better visualize the position of the networks with respect to the spectral class representatives, we performed a two-dimensional embedding of the spectral densities of all graphs. The embedding has been achieved by uniformly sampling the spectral density of each graph and by transforming the resulting space with a principal component analysis. We retain the first two principal components, which account for $\sim 89\%$ of the total variance. The results are shown in Fig. 6.3. Each data point is a network belonging to a class and the large filled dots represent the spectral class centres. The radius of the dashed circles represents the compactness of the cluster and corresponds to the diagonal elements of $d_{u,v}$.

Table 6.1. The diagonal elements correspond to the mean distance between each network and their own class' SCD, so they measure the compactness of the cluster. The off-diagonal elements are instead a measure of the clusters separation, each one being the mean distance of networks of a given class with respect to another class' SCD.

×	**PCN** (SCD)	**Bartoli** (SCD)	**LMGRS** (SCD)
PCN	0.2389	0.8483	0.7394
Bartoli	0.8540	0.2541	0.4278
LMGRS	0.7551	0.4376	0.2762

Looking at Fig. 6.3, it is evident how real PCNs occupy a different region of the space with respect to simulated models. This points to the need to fill this gap with additional refinements. Notably, we complement the *a priori* approach by exploiting a purely data-driven empirical procedure.

We tested the genetic algorithm reconfiguration method herein developed by performing the optimization for 10 different graphs, with number of vertices/edges randomly selected among the real values taken from the PCN set. In the following, we denote these new networks as the LMGRS-GEN ensemble.

As a first test, the weighting function of the distance $\tilde{d}(\cdot,\cdot)$ in Eq. (6.7) was set as a constant. Even if this choice seems intuitively the most unbiased, it does not lead to an effective optimization

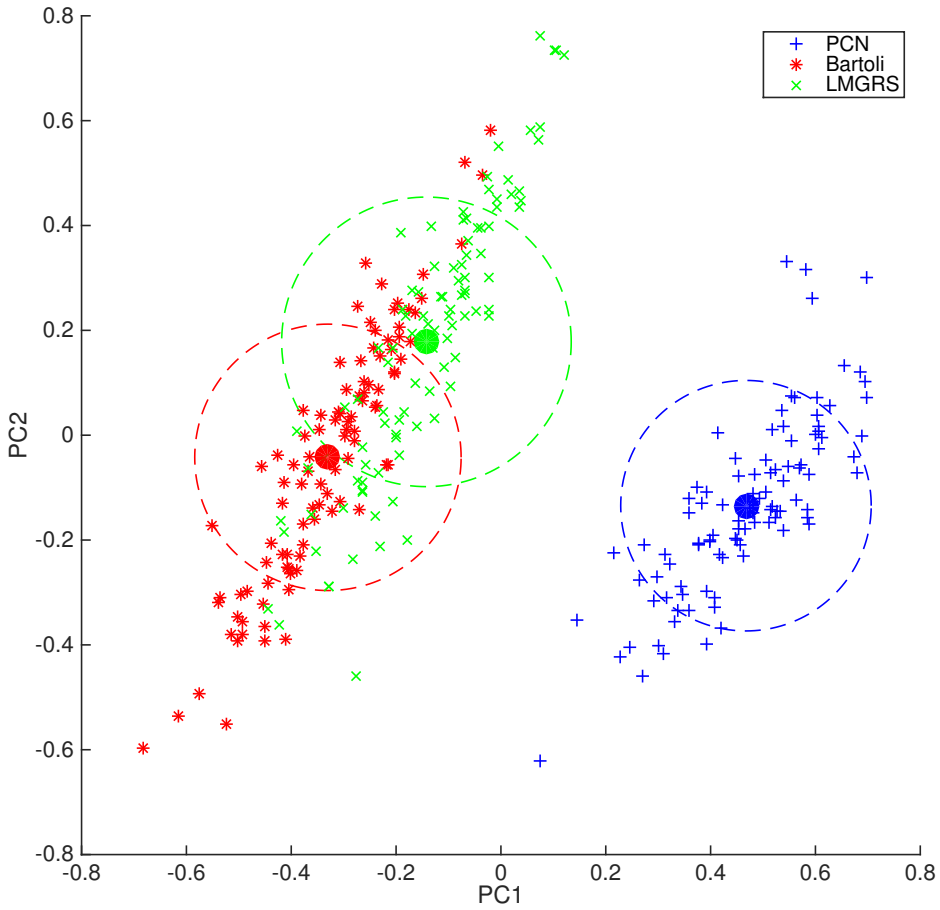


Figure 6.3. Colors online. Each point is a network and the large filled dots are the spectral class centers. The radius of the dashed circles represents the compactness of the cluster and corresponds to the diagonal elements of $d_{u,v}$. Notice the overlap between Bartoli, and LMGRS, which are generated with fairly similar processes.

of the spectrum in reasonable computing time. In fact, with such a weighting we observed that the reconfiguration is strongly biased in optimizing the rightmost part of the spectrum of the target density, i.e., the higher eigenvalues. The leftmost part, instead, is not satisfyingly optimized (details not shown here). A reason that explains this behavior lies in the structure of the normalized Laplacian spectrum. It is well-known that the leftmost part of the normalized Laplacian spectrum is related to the global, modular organization of the network [122]. In particular, the magnitude of the first nonzero eigenvalue, the so-called *spectral gap*, gives information on the separation of different modules and many other properties related to diffusion and synchronization processes [107]. It is hence reasonable to assume that the left-most side of the spectrum controls mostly the global features of the graph topology, while the right-most part is related to the local traits. Therefore, the right-most part of the spectrum should be considered as "easier" to reproduce. Indeed, the same variation on different eigenvalues can have significantly different effects on the network's global structure, depending on how much they differ in rank in the spectrum. In these cases, a weighting function is a convenient way to encode in the algorithm *a priori* information on the relative importance of different sections of

Table 6.2. Matrix of distances of LMGRS-GEN networks with respect to the PCN class and β values of the corresponding NNSD.

ID	N	E	Distance	β
M385	385	1573	0.0421	1.0499
M400	400	1583	0.0404	1.0786
M428	428	1683	0.0350	0.8696
M445	445	1794	0.0338	0.8905
M509	509	2039	0.0368	0.8988
M509	547	2181	0.0370	0.8388
M600	600	2382	0.0395	0.9389
M702	702	2775	0.0341	1.1109
M811	811	3215	0.0294	0.9485
M938	938	3706	0.0323	0.8288

the spectrum and to establish different priorities in the optimization. For this reason, in our work we have used exponentially decaying weights in Eq. (6.7), as depicted in Fig. 6.4. In the figure is shown the SCD of PCN alongside the weighting function $w(x)$.

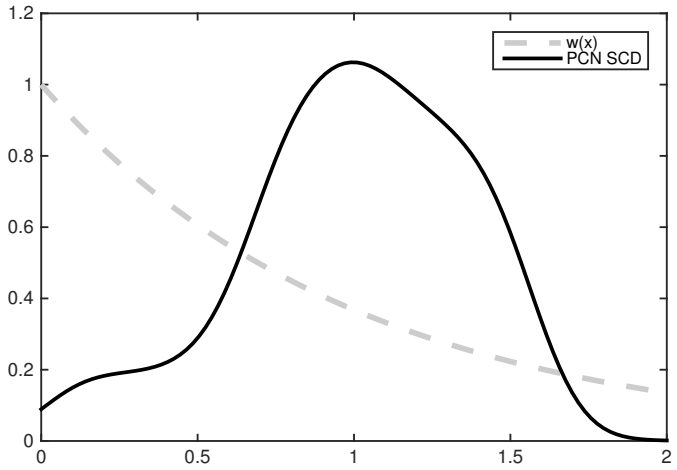


Figure 6.4. Weight function $w(x)$ (gray dashed line) of the distance defined in Eq. (6.7). The solid line is the SCD of PCN.

The genetic algorithm has been executed in all experiments described in Sec. 6.4 with the parameters $N_{\text{pop}} = 100$, $\mu_{\text{mut}} = 0.05$, $\mu_{\text{p}} = 0.01$, $\mu_{\text{cross}} = 1$, $n_{\text{sel}} = 50$.

The results of the optimization are shown in Table 6.2. For each pair (N,E) of graph dimension and connectivity values, the spectral distances obtained are very small with respect to the mean distances shown in Table 6.1. Notice that while the optimization has been carried out with the weighted objective function described by (6.7), the distances shown in Table 6.2 are calculated with the original unweighted distance of eq. (6.3), and they are considerably lower than the initial mean distances of the original populations of LMGRS networks. This confirms that the optimization of the weighted distance (6.7) has led to a successful optimization of the unweighted distance (6.4) as well. It is also important to point out that the chosen values for (N,E) are not guided by some constraint and, in principle, our reconstruction method allows to generate (i.e., reconfigure) graphs of arbitrary sizes. The meaning of the column indexed with β will be explained later.

In Fig. 6.5 we show a comparison of the spectral density of one of the LMGRS-GEN graphs,

namely M385 (385 vertices and 1576 edges), with respect to all SCDs under consideration. It is possible to notice a significant improvement in accuracy with respect to the SCD of LMGRS networks. In fact, the graph spectral density of M385 is nearly indistinguishable from the SCD of PCNs. It is worth focusing the attention on the left part of the spectrum, corresponding to the modular organization of the network, which is a fundamental property of the protein structure. The results about other LMGRS-GEN networks are similar and are not shown here only for the sake of brevity.

In Fig. 6.6 it is possible to observe a sample of an adjacency matrix with 385 vertices and 1576 edges taken from each ensemble. As it is possible to notice even by a visual inspection, the genetic algorithm based reconfiguration produces a more realistic matrix with respect to Bartoli and LMGRS. Even if there are no evident secondary structure elements, it is possible to recognize some common features between PCN and LMGRS-GEN in the coarse-grained organization of edges. Instead, the LMGRS model captures only the prior distribution of edges as a function of the backbone distance. In Fig. 6.7 are shown the distributions of

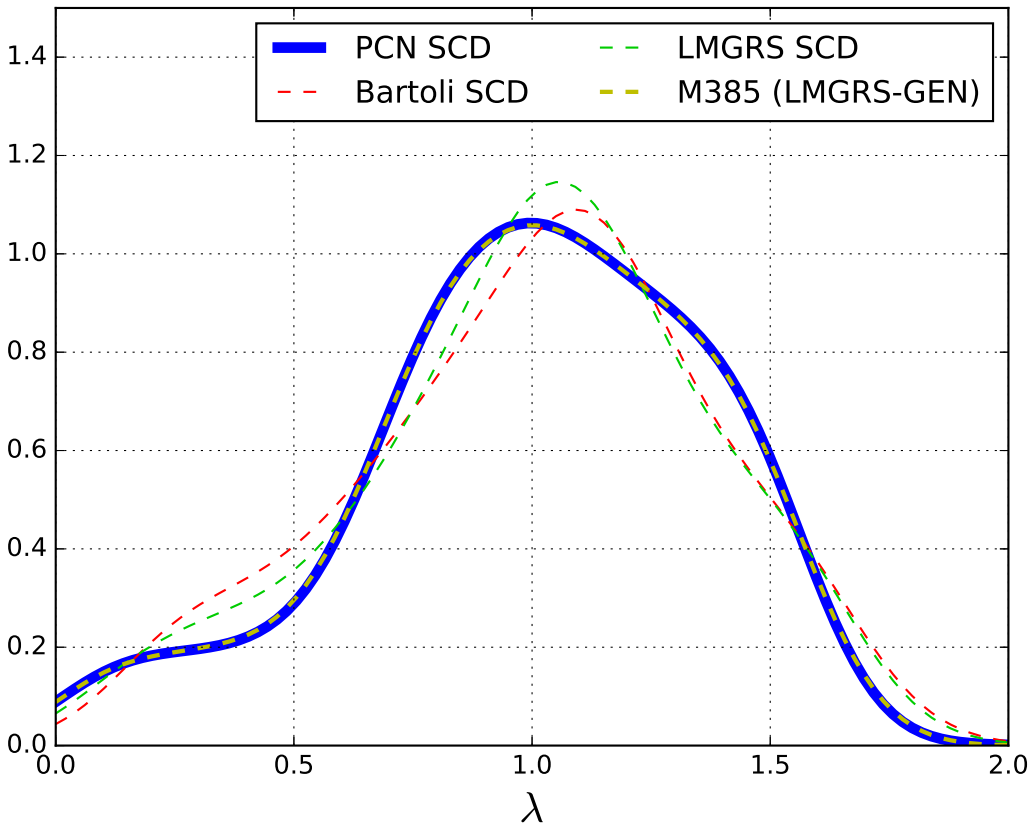


Figure 6.5. Colors online. Comparison of SCD for PCN (thick, blue), Bartoli (red, dashed), LMGRS (green, dashed), and the spectral density of M385 of LMGRS-GEN (yellow, dashed). The SCD of PCN and the spectral density of M385 are nearly identical.

several topological properties of LMGRS-GEN with respect to the LMGRS and PCN networks, specifically the average clustering coefficient (ACC), the degree assortativity coefficient (DAC), the average shortest path (ASP) and the modularity (MOD). As it is possible to observe, LMGRS-GEN are much more similar to PCN in terms of average clustering coefficient and modularity with respect to LMGRS. The improvement in similarity of these two properties can be explained

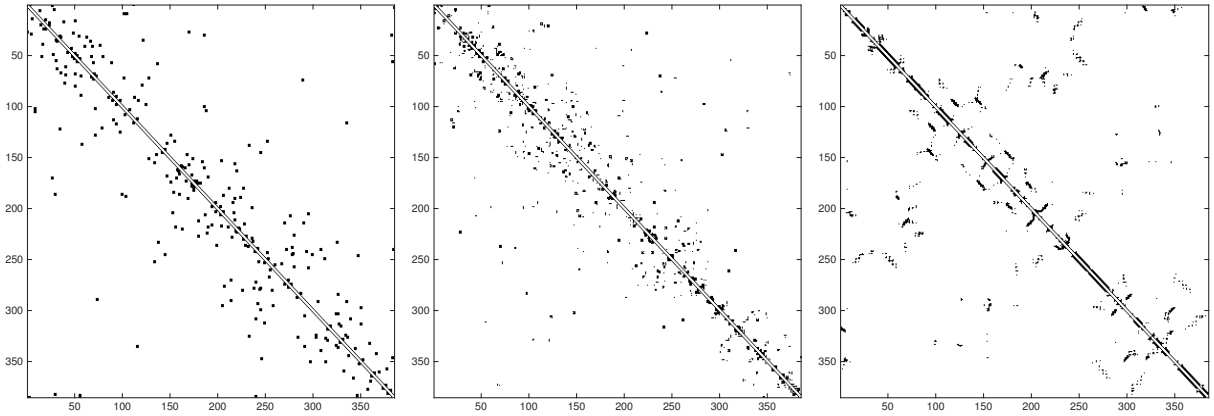


Figure 6.6. Comparison of the adjacency matrices of LMGRS (left), LMGRS-GEN (middle), and PCN (right). Notice that LMGRS and LMGRS-GEN are not designed to be the exact structural reconstruction of the PCN shown on the right. In fact, they are conceived to statistically approximate the spectral properties of a typical PCN.

by considering that the modular organization and the distribution of triangles in a network are strictly related to the spectral properties of its corresponding normalized laplacian [16, 138]. Conversely, LMGRS-GEN networks do not show significant improvement in terms of degree assortativity and average shortest path. This discrepancy can be interpreted by considering the fact that the average shortest path and the degree assortativity coefficient are properties that are significantly related to the underlying tridimensional nature of proteins, a feature that is not directly enforced in the optimization. In fact, since protein contact networks have to satisfy physical constraints given by the spatial dimensions of residues, many possible network configurations are not allowed and the average shortest path is generally longer. In the same way a higher assortativity, corresponding to the tendency of nodes with similar degree to be connected, is expected in 3D structures because of the correspondence between node degree and local residue density. Areas with higher local density of residues, in fact, will correspond to groups of high degree nodes connected with each other, while the opposite holds for low density areas.

Let us now discuss the results obtained for LMGRS-GEN in terms of RMT. Even if the spectral density captures many important structural and dynamic aspects of a network, another important role in defining the global organization is played by the correlations between the eigenvalues [88]. In fact, as confirmed by the Wigner's semicircle law, networks with significantly different structures can lead to the same spectral class distribution [118]. Accordingly, we have analyzed the correlation properties of the LMGRS-GEN spectra with respect to PCN. We have calculated the NNSD for each single LMGRS-GEN and PCN network as well as the ensemble NNSD of PCN. The obtained spacing distributions are then fitted with the Brody formula (6.6). The results of the calculated βs are shown, as a function of network size, in Fig. 6.8. The value of $\beta_{\mathrm{PCN}} = 0.8826$ obtained for the PCN ensemble denotes more consistency with the GOE ensemble. The β values oscillate between 0.7 and 1.1, so the different networks in the PCN ensemble have similar correlation properties, which are independent with the size of the network. The spread of β on the domain is reasonable, given the structural and functional diversity of the chosen PCNs of the original dataset [106]. The NNSD of LMGRS-GEN networks yields the β values reported in the last column of Table 6.2. These values are well-embedded in the PCN bulk, so we can conclude that also the correlation properties of the LMGRS-GEN eigenvalues are similar to those characterizing real PCNs. Finally, in order to numerically justify the method presented in Sec. 6.1.1 to calculate the NNSD of an ensemble containing matrices

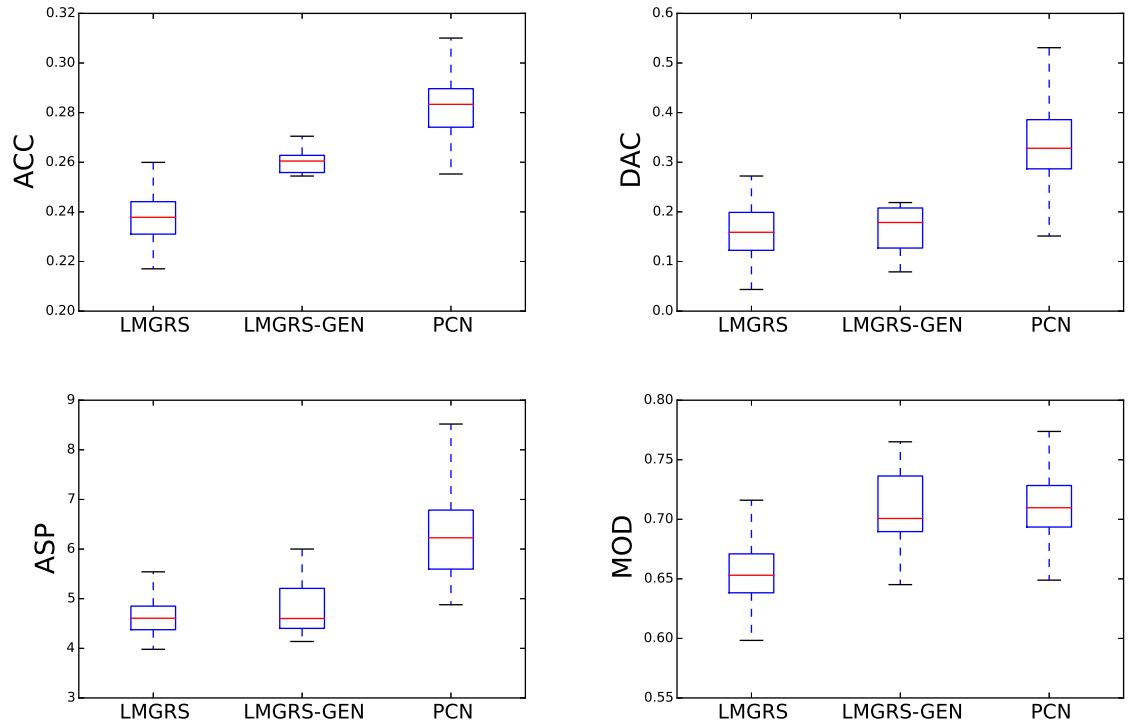


Figure 6.7. Colors online. Boxplots of the distributions of the topological properties of the LMGRS-GEN with respect to LMGRS and PCN. The red marker represents the median value, the box is bounded by the 1st and 3rd quartiles and the whiskers' extent is 1.5 × IQR where IQR is the interquartile range.

of different sizes, we have also calculated the mean of the β values of PCN and compared it to β_{PCN}. The outcome is $\bar{\beta}_{\mathrm{PCN}} = 0.8826$, which is equal to β_{PCN}, so we conclude that the result for the ensemble NNSD is consistent. These results suggest a further analysis of the residual differences in the spectra also in terms of long-range correlations, for instance by evaluating their Δ_3 statistics [118].

6.5 Discussion

In this Chapter, we have analyzed the structural organization of protein contact networks by focusing the analysis on their normalized Laplacian spectra. Starting from an a priori generative model of a typical protein contact map, we developed an optimization method based on genetic algorithms that rewires the edges of the synthetic map in order to optimize the similarity of their normalized Laplacian spectrum distribution with the spectral distribution of the ensemble of all available protein contact networks. This problem is similar to the inverse spectral problem, with the important difference that the target spectrum that we wanted to reproduce is an average kernel-estimated density. Therefore, in practice it does not correspond to any specific graph of the dataset, but it is simultaneously similar to every spectrum of the class, assuming homogeneity among the constituting spectral densities. This allows us to investigate properties that are common among all the considered proteins, while washing out the details and statistical fluctuations of the single elements. The reconfigured networks generated by the proposed procedure, LMGRS-GEN, are significantly closer, in terms of spectra and several topological properties, to the real protein contact networks, while their adjacency matrix does not clearly show secondary structure elements. This raises the question of whether the new networks act as

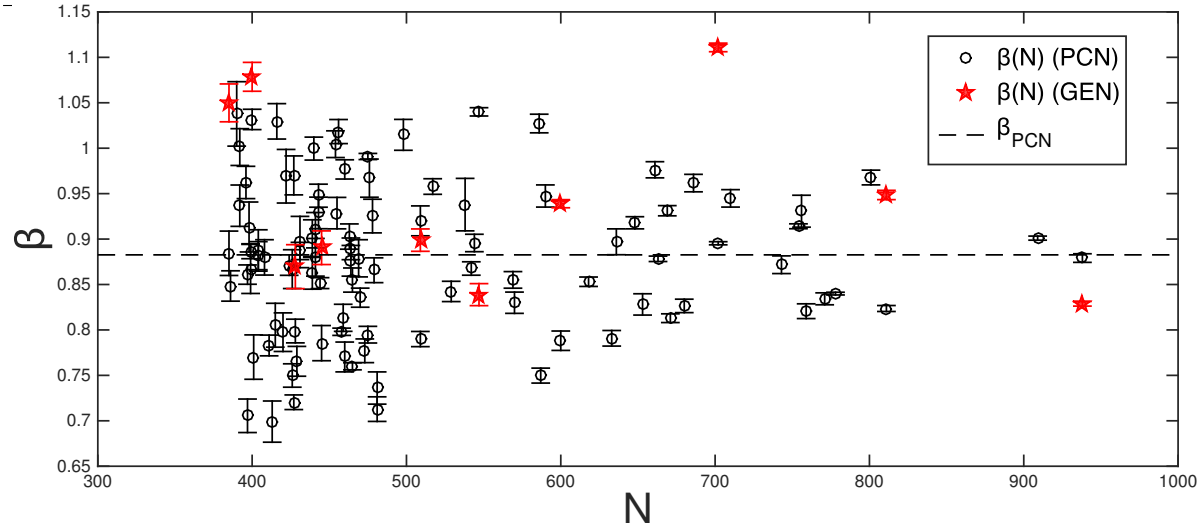


Figure 6.8. Colors online. Calculated values of β (6.6) for single matrices of PCN and LMGRS-GEN networks (black circles and red stars, respectively) and value of β_{PCN} (dashed line). The error bar is the mean-square error of the fit and the dashed line represents β_{PCN}.

a sort of "effective model", which reproduces the spectral properties of the real proteins while still being different in some aspects of the structural organization. This is reasonable and to some extent expected, since real proteins have to satisfy physical constraints that significantly reduce the space of realizable contact maps [55, 166]. In this work, however, we are interested in basic topological design principles, without focusing on the physical substrate of a protein. As a consequence, physical constraints have not been explicitly included in the optimization process and the result is that LMGRS-GEN networks have more degrees of freedom in the possible configurations that they can assume. Arguably, each of these networks represents an "average topology", built according to design principles encoded implicitly in the averaged Laplacian spectrum density – i.e. the SCD of PCNs – and common to all the considered PCNs. According to this point of view, high-level features, like secondary structure elements, are filtered out as being part of the particular characteristics of single networks/proteins. This is supported by the fact that in every protein the number, dimension, and position of α-helices and β-sheets can vary significantly. Another point to consider is the fact that these networks have been optimized in order to resemble only the static functional form of the PCN spectral class density, without explicitly taking into account the correlations between single eigenvalues in the spectrum. Even though the results in Sec. 6.4 already show a good agreement in first-neighbor correlations between PCN and LMGRS-GEN, an interesting development in this direction would be to design an improved optimization problem in which higher-order and/or long-range correlations between eigenvalues are taken into account. In conclusion, the future directions for this work are several, notably (i) the inclusion of realizability criteria in the generation/optimization of adjacency matrices in order to satisfy the physical constraints of real proteins; (ii) the analysis of the impact of higher-order and long-range correlations between eigenvalues in the global organization of proteins via random matrix theory. While the first direction is more concerned with protein physical properties, in the second case the problem is approached from a purely abstract point of view. This allows for the study of the universal properties of the normalized Laplacian of a graph and its role in the generation of complexity, which could then be applied to other fields where the structural organization and dynamical properties of a network are central issues in the understanding of the system under consideration.

Appendix A

Echo State Networks

ESNs belong to the class of computational dynamical systems, implemented according to the biologically-inspired reservoir computing approach [112]. An input signal is fed to a large, recurrent and randomly connected hidden layer, the reservoir, whose outputs are combined by a memory-less linear layer, called readout, to solve a specified task. ESNs have been adopted in a variety of different contexts, such as time series prediction [26], static classification [6], speech recognition [154], adaptive control [75] harmonic distortion measurements [115] and, in general, for modeling of various kinds of non-linear dynamical systems [74]. A schematic depiction of an ESN is shown in Fig. A.1.

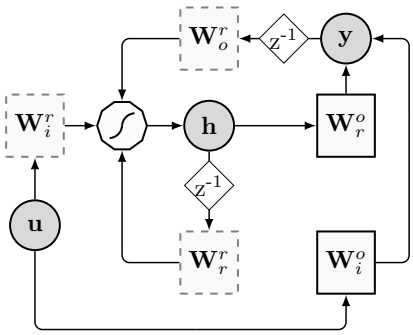


Figure A.1. Schematic depiction of an ESN. The circles represent the input variables $\mathbf{u}$, the state variables $\mathbf{h}$ and the output variables $\mathbf{y}$. The squares depicted with solid lines, $\mathbf{W}_r^o$ and $\mathbf{W}_i^o$, are the trainable weight matrices of the readout, while the squares with dashed lines, $\mathbf{W}_r^r$, $\mathbf{W}_o^r$ and $\mathbf{W}_i^r$, are random initialized weight matrices. The polygon represents the non-linear transformation performed by neurons and z^{-1} is the backshift/lag operator.

The current output of an ESN is computed in two distinct phases. First, the N_i-dimensional input vector $\mathbf{u}(t) \in \mathbb{R}^{N_i}$ is given as input to the recurrent reservoir, whose internal state $\mathbf{h}(t-1) \in \mathbb{R}^{N_r}$ is updated according to the state equation:

$$\mathbf{h}(t) = f_{\text{res}}\left(\mathbf{W}_i^r \mathbf{u}(t) + \mathbf{W}_r^r \mathbf{h}(t-1) + \mathbf{W}_o^r \mathbf{y}(t-1)\right), \tag{A.1}$$

where $\mathbf{W}_i^r \in \mathbb{R}^{N_r \times N_i}$, $\mathbf{W}_r^r \in \mathbb{R}^{N_r \times N_r}$ and $\mathbf{W}_o^r \in \mathbb{R}^{N_r \times N_o}$ are randomly initialized at the beginning of the learning process, and they remain unaltered afterwards. $f_{\text{res}}(\cdot)$ in Eq. (A.1) is a suitable non-linear function, typically a sigmoid, and $\mathbf{y}(t-1) \in \mathbb{R}^{N_o}$ is the previous output of the network. In our case, we have $f_{\text{res}}(\cdot) = \tanh(\cdot)$. In the second phase, the ESN prediction is computed according to:

$$\mathbf{y}(t) = \mathbf{W}_i^o \mathbf{u}(t) + \mathbf{W}_r^o \mathbf{h}(t), \tag{A.2}$$

where $\mathbf{W}_i^o \in \mathbb{R}^{N_o \times N_i}$, $\mathbf{W}_r^o \in \mathbb{R}^{N_o \times N_r}$ are trainable connections. The difference between fixed and adaptable weight matrices is shown in Fig. A.1 with the use of continuous and dashed lines, respectively.

Finally, a few words should be spent on the choice of the matrix $\mathbf{W}_r^r$. According to the ESN theory, the reservoir must satisfies the so-called "echo state property" (ESP) [112]. This means

that the effect of a given input on the state of the reservoir must vanish in a finite number of time-instants. In this paper we adopt the widely used rule-of-thumb that suggests to rescale the matrix $\mathbf{W}_r^r$ to have $\rho(\mathbf{W}_r^r) < 1$, where $\rho(\cdot)$ denotes the spectral radius.

To determine the weight matrices of the readout, let us consider a training sequence of T_{tr} desired input-outputs pairs $\{\mathbf{u}(t), \mathbf{d}(t)\}_{t=1}^{T_{\text{tr}}}$, where the output is given by $\mathbf{d}(t) = \mathbf{u}(t + \tau_f)$. Here, τ_f defines the forecast horizon (or step ahead) considered in the prediction, i.e. how far ahead in time the input signal must be predicted. In the initial phase of training, called "state harvesting", the inputs are fed to the reservoir in accordance with Eq. (A.1), producing a sequence of internal states $\{\mathbf{h}(t)\}_{t=1}^{T_{\text{tr}}}$. Since, by definition, the outputs of the ESN are not available for feedback, the desired output is used instead in Eq. (A.2) (the so-called "teacher forcing"). The states are stacked in a matrix $\mathbf{H} \in \mathbb{R}^{T_{\text{tr}} \times N_i + N_r}$ and the desired outputs in a vector $\mathbf{d} \in \mathbb{R}^Q$:

$$\mathbf{H} = \begin{bmatrix} \mathbf{u}^T(1), \mathbf{h}^T(1) \\ \vdots \\ \mathbf{u}^T(T_{\text{tr}}), \mathbf{h}^T(T_{\text{tr}}) \end{bmatrix}, \tag{A.3}$$

$$\mathbf{d} = \begin{bmatrix} d(1) \\ \vdots \\ d(T_{\text{tr}}) \end{bmatrix}. \tag{A.4}$$

The initial D rows from Eq. (A.3) and Eq. (A.4) should be discarded, since they refer to a transient phase in the ESN's behavior. We refer to them as the washout elements.

At this point the resulting training problem is a standard linear regression, which can be solved in a large variety of ways. We used the least-square regression, which is the algorithm originally proposed for training the readout [85]. It consists in the following regularized least-square problem:

$$\mathbf{w}_{\text{ls}}^* = \underset{\mathbf{w} \in \mathbb{R}^{N_i + N_r}}{\arg\min} \frac{1}{2} \|\mathbf{H}\mathbf{w} - \mathbf{d}\|_2^2 + \frac{\alpha}{2} \|\mathbf{w}\|_2^2 , \tag{A.5}$$

where $\mathbf{w} = [\mathbf{w}_i^o \, \mathbf{w}_r^o]^T$ and $\alpha \in \mathbb{R}^+$ is a positive scalar known as *regularization factor*. A solution of problem (A.5) can be obtained in closed form as:

$$\mathbf{w}_{\text{ls}}^* = \left(\mathbf{H}^T\mathbf{H} + \alpha\mathbf{I}\right)^{-1} \mathbf{H}^T\mathbf{d}. \tag{A.6}$$

Whenever $N_r + N_i > Q$, Eq. (A.6) can be computed more efficiently by rewriting it as:

$$\mathbf{w}_{\text{ls}}^* = \mathbf{H}^T \left(\mathbf{H}\mathbf{H}^T + \alpha\mathbf{I}\right)^{-1} \mathbf{d}. \tag{A.7}$$

Once the readout layer is trained, when the network is fed with an unseen input signal $\mathbf{u}(t)$, with $t > T_{\text{tr}}$, it returns a predicted value $\hat{\mathbf{y}}(t) = \mathbf{u}(t + \tau_f)$, according to the step ahead τ_f defined in the training phase.

Conclusions

In this thesis we explored the organization principles between protein structure by means of their network representation. By employing several techniques of graph theory, computational intelligence and machine learning we showed that Protein Contact Networks have many peculiarities that distinguish them from other kinds of networks. The normalized graph Laplacian representation, related to the properties of connectivity and diffusion along the networks edges, is at the heart of this analysis. The study is structured in two main phases. An analysis phase where we employed several graph-theoretic tools to investigate the structural properties of Protein Contact Networks and compare them with several other biological networks, and a generation phase, where we designed a generative model capable of generating networks that present such features. In the first phase we extracted the heat kernel operator as the solution of a first order differential equation dependent of the Laplacian. From the heat kernel we obtained several graph invariants related to the heat diffusion on the graph, namely the heat trace, heat content and the heat content invariant coefficients. By embedding these quantities in a suitable vector space composed by several groups of biological and synthetic network we were able to observe the descriptive capability of the considered heat kernel invariants in distinguishing different kinds of networks. By means of a Canonical Correlation Analysis we measured an agreement in description between the heat kernel spaces and the space of topological features, directly extracted from the networks. This allowed us to consider the heat kernel invariants as an indirect yet meaningful representation of the main topological features of PCN. We then defined the concept of spectral ensembles, i.e. groups of network with similar spectral distributions, and the ensemble heat trace as a function of network size for a fixed time. By studying the linear fitting slopes of the ensemble heat trace we obtained a characterization of the heat trace decay of the whole ensemble, from which we observed that heat diffusion on PCN is described by two different regimes. After a first phase of normal diffusion, for longer times the heat presents a subdiffusive behavior. This graph-theoretic observation is supported by experimental measurements on energy flow and vibrational dynamics. Moreover, this property is not observed on any other analyzed network, including the network generated with the scheme proposed in Bartoli et al. [20]. This result highlights a considerable difference in wiring organization between Protein Contact Networks and the Bartoli networks. The second part of the analysis regards the study of random walks on networks. In this part we studied PCN structure from a different viewpoint. We setup a random walk process on the graph's topology and at each time we evaluated several local node observables, namely vertex degree, vertex clustering coefficient and vertex closeness centrality. We then proceeded to analyze these time series by means of Multifractal Detrended Fluctuation Analysis, in order to find the trace of long term correlations in data. From the results we obtain that for PCN the time series of all observables are persistent, indicating a strong assortativity of the corresponding network. On the other hand, also in this case the synthetic networks from the Bartoli ensemble do not capture this persistence property. Successively, we presented a methodology to perform a detrending of time series in a data-driven way, by employing Echo State Networks. In several

synthetic and real-world tests the technique showed state-of-art performances in filtering the nonlinearities from the time series. We then proceeded to the second phase about the generation of statistically realistic protein contact networks. In a first work, we proposed a variant of the Bartoli model. In our scheme, the connection probability between two residues is evaluated from the empirical frequencies observed in a dataset of real Protein Contact Networks. With this modification the new generated LMGRS networks present an improved similarity both in heat trace and in spectral distribution to the real PCN. They show an increased subdiffusive character with respect to the Bartoli networks, even if to a lesser extent with respect to PCN. We also evaluated the spectral distributions of LMGRS, Bartoli and PCN and found LMGRS closer to PCN, especially in the left part of the spectrum. In further analysis we measured a discrepancy in the value of the average shortest path of LMGRS with respect to PCN. To alleviate this difference, we performed a reconfiguration step aimed at decreasing the small world character of LMGRS, obtaining the LMGRS-REC ensemble. In a last work, we set an optimization problem. The objective of the optimization is to obtain networks with a spectral distribution identical to the one of PCN. This objective is achieved with the use of a genetic algorithm, equipped with custom-designed operators. Further analyses on the topological properties of the generated networks, LMGRS-GEN, revealed that the new networks have increased similarity to PCN in terms of modularity and average clustering coefficient.

In these works we showed how the descriptive power of the network representation and the versatility of computational intelligence techniques allow to gain considerable insights on protein structure, without considering further chemical details. The improvement of such methodologies is a key in tackling problems as complex as protein folding and to pave the way for the discovery of universal principles at the basis of biological organization.

Bibliography

[1] DREAM5. URL `http://wiki.c2b2.columbia.edu/dream/index.php/D5c3`.

[2] Protein Data Bank. URL `http://www.rcsb.org/pdb/home/home.do`.

[3] Daily sunspot number. URL `http://www.sidc.be/silso/datafiles`. last accessed on 01-Jul-2015.

[4] S M. Abuelenin and A Y. Abul-Magd. Effect of unfolding on the spectral statistics of adjacency matrices of complex networks. *Procedia Computer Science*, 12:69–74, 2012. doi: 10.1016/j.procs.2012.09.031.

[5] Réka Albert and Albert-László Barabási. Statistical mechanics of complex networks. *Reviews of modern physics*, 74(1):47, 2002.

[6] Luís A Alexandre, Mark J Embrechts, and Jonathan Linton. Benchmarking reservoir computing on time-independent classification tasks. In *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*, pages 89–93. IEEE, 2009.

[7] Benjamin Amor, S N Yaliraki, Rudiger Woscholski, and Mauricio Barahona. Uncovering allosteric pathways in caspase-1 with markov transient analysis and multiscale community detection. *Molecular BioSystems*, 10:2247–2258, 2014.

[8] Philip W Anderson et al. More is different. *Science*, 177(4047):393–396, 1972.

[9] William N Anderson Jr and Thomas D Morley. Eigenvalues of the laplacian of a graph. *Linear and multilinear algebra*, 18(2):141–145, 1985.

[10] M. Ausloos. Generalized hurst exponent and multifractal function of original and translated texts mapped into frequency and length time series. *Physical Review E*, 86:031108, Sep 2012. doi: 10.1103/PhysRevE.86.031108.

[11] Sivaraman Balakrishnan, Hetunandan Kamisetty, J G Carbonell, S-I Lee, and C J Langmead. Learning generative models for protein fold families. *Proteins: Structure, Function, and Bioinformatics*, 79(4):1061–1078, 2011. doi: 10.1002/prot.22934.

[12] Carlo Baldassi, Marco Zamparo, Christoph Feinauer, Andrea Procaccini, Riccardo Zecchina, Martin Weigt, and Andrea Pagnani. Fast and accurate multivariate Gaussian modeling of protein families: Predicting residue contacts and protein-interaction partners. *PLoS ONE*, 9(3):e92721, 2014. doi: 10.1371/journal.pone.0092721.

[13] J R Banavar and Saraswathi Vishveshwara. Protein structure and folding–simplicity within complexity. *Journal of Biomolecular Structure and Dynamics*, 31(9):973–975, 2013. doi: 10.1080/07391102.2012.748533.

[14] Anirban Banerjee and Jürgen Jost. Laplacian spectrum and protein-protein interaction networks. *ArXiv eprints*, page 7, 2007. URL `http://arxiv.org/abs/0705.3373`.

[15] Anirban Banerjee and Jürgen Jost. Spectral plots and the representation and interpretation of biological data. *Theory in Biosciences*, 126(1):15–21, 2007. ISSN 14317613. doi: 10.1007/s12064-007-0005-9.

[16] Anirban Banerjee and Jürgen Jost. On the spectrum of the normalized graph laplacian. *Linear Algebra and Its Applications*, 428(11-12):3015–3022, 2008. ISSN 00243795. doi: 10.1016/j.laa.2008.01.029.

[17] Anirban Banerji and Indira Ghosh. Fractal symmetry of protein interior: what have we learned? *Cellular and Molecular Life Sciences*, 68(16):2711–2737, 2011.

[18] Albert-László Barabási and Réka Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.

[19] Albert-Laszlo Barabasi and Z N Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.

[20] L Bartoli, P Fariselli, and R Casadio. The effect of backbone on the small-world properties of protein contact maps. *Physical biology*, 4(4):L1, 2007.

[21] Jozef Barunik and Ladislav Kristoufek. On hurst exponent estimation under heavy-tailed distributions. *Physica A: Statistical Mechanics and its Applications*, 389(18):3844–3855, 2010.

[22] Jozef Barunik, Tomaso Aste, Tiziana Di Matteo, and Ruipeng Liu. Understanding the source of multifractality in financial markets. *Physica A: Statistical Mechanics and its Applications*, 391(17):4234–4251, 2012.

[23] Amir Bashan, Ronny Bartsch, J W Kantelhardt, and Shlomo Havlin. Comparison of detrending methods for fluctuation analysis. *Physica A: Statistical Mechanics and its Applications*, 387(21):5080–5090, 2008.

[24] William Bialek, Ilya Nemenman, and Naftali Tishby. Predictability, complexity, and learning. *Neural Computation*, 13(11):2409–2463, 2001. doi: 10.1162/089976601753195969.

[25] F M Bianchi, Simone Scardapane, Aurelio Uncini, Antonello Rizzi, and Alireza Sadeghian. Prediction of telephone calls load using echo state network with exogenous variables. *Neural Networks*, 71:204–2013, 2015. doi: 10.1016/j.neunet.2015.08.010.

[26] F.M. Bianchi, E. De Santis, A. Rizzi, and A. Sadeghian. Short-term electric load forecasting using echo state networks and PCA decomposition. *Access, IEEE*, PP(99):1–1, 2015. ISSN 2169-3536. doi: 10.1109/ACCESS.2015.2485943.

[27] V D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008, 2008.

[28] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D. Hwang. Complex networks: Structure and dynamics. *Physics Reports*, 424(4-5):175–308, Feb. 2006. ISSN 03701573. doi: 10.1016/j.physrep.2005.10.009.

[29] Csaba Böde, I A Kovács, M S Szalay, Robin Palotai, Tamás Korcsmáros, and Péter Csermely. Network analysis of protein dynamics. *Febs Letters*, 581(15):2776–2782, 2007.

[30] Moreno Bonaventura, Vincenzo Nicosia, and Vito Latora. Characteristic times of biased random walks on complex networks. *Physical Review E*, 89(1):012803, 2014.

[31] Wouter Boomsma, K V Mardia, C C Taylor, Jesper Ferkinghoff-Borg, Anders Krogh, and Thomas Hamelryck. A generative, probabilistic model of local protein structure. *Proceedings of the National Academy of Sciences*, 105(26):8932–8937, 2008. doi: 10.1073/pnas.0801715105.

[32] Luis Boza and MP Revuelta. The dimension of a graph. *Electronic Notes in Discrete Mathematics*, 28:231–238, 2007.

[33] Raffaella Burioni and Davide Cassi. Random walks on graphs: ideas, techniques and results. *Journal of Physics A: Mathematical and General*, 38(8):R45, 2005. doi: 10.1088/0305-4470/38/8/R01.

[34] Sanjeev Chauhan, Michelle Girvan, and Edward Ott. Spectral properties of networks with community structure. *Physical Review E*, 80(5):056114, 2009. doi: 10.1103/PhysRevE.80.056114.

[35] Ashvin Chhabra and R V. Jensen. Direct determination of the f(α) singularity spectrum. *Physical Review Letters*, 62:1327–1330, Mar 1989. doi: 10.1103/PhysRevLett.62.1327.

[36] C V Chianca, A Ticona, and T J P Penna. Fourier-detrended fluctuation analysis. *Physica A: Statistical Mechanics and its Applications*, 357(3):447–454, 2005. doi: 10.1016/j.physa.2005.03.047.

[37] Fabrizio Chiti, Niccolò Taddei, P M White, Monica Bucciantini, Francesca Magherini, Massimo Stefani, and C M Dobson. Mutational analysis of acylphosphatase suggests the importance of topology and contact order in protein folding. *Nature Structural & Molecular Biology*, 6(11):1005–1009, 1999. doi: 10.1038/14890.

[38] Fan Chung. The heat kernel as the pagerank of a graph. *Proceedings of the National Academy of Sciences*, 104(50):19735–19740, 2007. doi: 10.1073/pnas.0708838104.

[39] Francesc Comellas and Jordi Diaz-Lopez. Spectral reconstruction of complex networks. *Physica A: Statistical Mechanics and its Applications*, 387(25):6436–6442, Nov. 2008. doi: 10.1016/j.physa.2008.07.032.

[40] L da F Costa, F. A. Rodrigues, Gonzalo Travieso, and P R Villas Boas. Characterization of complex networks: A survey of measurements. *Advances in Physics*, 56(1):167–242, 2007.

[41] J P Crutchfield and D P Feldman. Regularities unseen, randomness observed: Levels of entropy convergence. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 13(1):25–54, 2003. doi: 10.1063/1.1530990.

[42] Andrew Currin, Neil Swainston, P J Day, and D B Kell. Synthetic biology for the directed evolution of protein biocatalysts: navigating sequence space intelligently. *Chemical Society Reviews*, pages –, 2015. doi: 10.1039/C4CS00351A.

[43] Joni Dambre, David Verstraeten, Benjamin Schrauwen, and Serge Massar. Information processing capacity of dynamical systems. *Scientific Reports*, 2, 2012. doi: 10.1038/srep00514.

[44] J. G. De Gooijer and R. J. Hyndman. 25 years of time series forecasting. *International Journal of Forecasting*, 22(3):443–473, 2006. doi: 10.1016/j.ijforecast.2006.01.001.

[45] David de Juan, Florencio Pazos, and Alfonso Valencia. Emerging methods in protein co-evolution. *Nature Reviews Genetics*, 14(4):249–261, 2013. doi: 10.1038/nrg3414.

[46] S C de Lange, M A de Reus, and M P van den Heuvel. The Laplacian spectrum of neural networks. *Frontiers in Computational Neuroscience*, 7, 2013. ISSN 1662-5188. doi: 10.3389/fncom.2013.00189. PMCID: PMC3888935.

[47] M. Dehmer and A. Mowshowitz. A history of graph entropy measures. *Information Sciences*, 181(1):57–78, 2011. ISSN 0020-0255. doi: 10.1016/j.ins.2010.08.041.

[48] J-C Delvenne, S N Yaliraki, and Mauricio Barahona. Stability of graph communities across time scales. *Proceedings of the National Academy of Sciences*, 107(29):12755–12760, 2010.

[49] L Di Paola, M De Ruvo, P Paci, D Santoni, and A Giuliani. Protein contact networks: an emerging paradigm in chemistry. *Chemical Reviews*, 113(3):1598–1613, 2012.

[50] L Di Paola, M De Ruvo, P Paci, D Santoni, and A Giuliani. Protein contact networks: an emerging paradigm in chemistry. *Chemical Reviews*, 113(3):1598–1613, 2012.

[51] Luisa Di Paola and Alessandro Giuliani. Protein contact network topology: a natural language for allostery. *Current Opinion in Structural Biology*, 31:43–48, 2015. doi: 10.1016/j.sbi.2015.03.001.

[52] Luisa Di Paola, Paola Paci, Daniele Santoni, Micol De Ruvo, and Alessandro Giuliani. Proteins as sponges: a statistical journey along protein structure organization principles. *Journal of chemical information and modeling*, 52(2):474–482, 2012.

[53] T G Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems: First International Workshop, MCS 2000 Cagliari, Italy, June 21–23, 2000 Proceedings*, pages 1–15. Springer-Verlag, London, UK, 2000. doi: 10.1007/3-540-45014-9_1.

[54] S Drożdż and P Oświęcimka. Detecting and interpreting distortions in hierarchical organization of complex time series. *Physical Review E*, 91(3):030902, 2015. doi: 10.1103/PhysRevE.91.030902.

[55] J M Duarte, Rajagopal Sathyapriya, Henning Stehr, Ioannis Filippis, and Michael Lappe. Optimal contact definition for reconstruction of contact maps. *BMC Bioinformatics*, 11(1):283, 2010. doi: 10.1186/1471-2105-11-283.

[56] X. Dutoit, B. Schrauwen, J. Van Campenhout, D. Stroobandt, H. Van Brussel, and M. Nuttin. Pruning and regularization in reservoir computing. *Neurocomputing*, 72(7):1534–1546, 2009.

[57] Magnus Ekeberg, Cecilia Lövkvist, Yueheng Lan, Martin Weigt, and Erik Aurell. Improved contact prediction in proteins: using pseudolikelihoods to infer potts models. *Physical Review E*, 87(1):012707, 2013. doi: 10.1103/PhysRevE.87.012707.

[58] M. B. Enright and D. M. Leitner. Mass fractal dimension and the compactness of proteins. *Physical Review E*, 71:011912, Jan 2005. doi: 10.1103/PhysRevE.71.011912.

[59] Ernesto Estrada. Universality in protein residue networks. *Biophysical Journal*, 98(5):890–900, 2010. doi: 10.1016/j.bpj.2009.11.017.

[60] Dustin Fetterhoff, Ioan Opris, S L Simpson, S A Deadwyler, R E Hampson, and R A Kraft. Multifractal analysis of information processing in hippocampal neural ensembles during working memory under Δ^9 tetrahydrocannabinol administration. *Journal of neuroscience methods*, 2014.

[61] Patrick Flandrin, Gabriel Rilling, and Paulo Goncalves. Empirical mode decomposition as a filter bank. *IEEE Signal Processing Letters*, 11(2):112–114, Feb. 2004. doi: 10.1109/LSP.2003.821662.

[62] R J Flassig, Sandra Heise, Kai Sundmacher, and Steffen Klamt. An effective framework for reconstructing gene regulatory networks from genetical genomics data. *Bioinformatics*, 29(2):246–254, 2013.

[63] Ruben Fossion, G T Vargas, and J C L Vieyra. Random-matrix spectra as a time series. *Physical Review E*, 88(6):060902, 2013. doi: 10.1103/PhysRevE.88.060902.

[64] L K Gallos, Chaoming Song, Shlomo Havlin, and Hernán A Makse. Scaling theory of transport in complex biological networks. *Proceedings of the National Academy of Sciences*, 104(19):7746–7751, 2007.

[65] L K Gallos, Chaoming Song, and Hernán A Makse. A review of fractality and self-similarity in complex networks. *Physica A: Statistical Mechanics and its Applications*, 386(2):686–691, 2007.

[66] Jianbo Gao, Yinhe Cao, W-W Tung, and Jing Hu. *Multiscale Analysis of Complex Time Series: Integration of Chaos and Random Fractal Theory, and Beyond*. John Wiley & Sons, New York, NY, USA, 2007.

[67] T F Gonzalez. *Handbook of Approximation Algorithms and Metaheuristics*. Chapman & Hall/CRC, London, UK, 2007. ISBN 1584885505.

[68] Rony Granek. Proteins as fractals: role of the hydrodynamic interaction. *Physical Review E*, 83(2):020902, 2011.

[69] Jiao Gu, Bobo Hua, and Shiping Liu. Spectral distances on graphs. *Discrete Applied Mathematics*, 190–191:56–74, 2015. ISSN 0166-218X. doi: 10.1016/j.dam.2015.04.011.

[70] Jiao Gu, Jürgen Jost, Shiping Liu, and Peter F. Stadler. Spectral classes of regular, random, and empirical graphs. *Linear Algebra and its Applications*, 489:30 – 49, 2016. ISSN 0024-3795. doi: 10.1016/j.laa.2015.08.038.

[71] Nabil Guelzim, Samuele Bottani, Paul Bourgine, and François Képès. Topological and causal structure of the yeast transcriptional regulatory network. *Nature genetics*, 31(1):60–63, 2002.

[72] Roger Guimera, Marta Sales-Pardo, and L A N Amaral. Modularity from fluctuations in random graphs and complex networks. *Physical Review E*, 70(2):025101, 2004. doi: 10.1103/PhysRevE.70.025101.

[73] Ivan Gutman and Bo Zhou. Laplacian energy of a graph. *Linear Algebra and its applications*, 414(1):29–37, 2006.

[74] Seong I Han and Jang M Lee. Fuzzy echo state neural networks and funnel dynamic surface control for prescribed performance of a nonlinear dynamic system. *Industrial Electronics, IEEE Transactions on*, 61(2):1099–1112, 2014.

[75] S.I. Han and J.M. Lee. Fuzzy echo state neural networks and funnel dynamic surface control for prescribed performance of a nonlinear dynamic system. *Industrial Electronics, IEEE Transactions on*, 61(2):1099–1112, Feb 2014. ISSN 0278-0046. doi: 10.1109/TIE.2013.2253072.

[76] L K Hansen and Peter Salamon. Neural network ensembles. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 12(10):993–1001, Oct. 1990. doi: 10.1109/34.58871.

[77] Leslie Hogben. Spectral graph theory and the inverse eigenvalue problem of a graph. *Electronic Journal of Linear Algebra*, 14(1):3, 2005.

[78] T A Hopf, L J Colwell, Robert Sheridan, Burkhard Rost, Chris Sander, and D S Marks. Three-dimensional structures of membrane proteins from genomic sequencing. *Cell*, 149 (7):1607–1621, 2012. doi: 10.1016/j.cell.2012.04.012.

[79] Jing Hu, Jianbo Gao, and Xingsong Wang. Multifractal analysis of sunspot time series: the effects of the 11-year cycle and Fourier truncation. *Journal of Statistical Mechanics: Theory and Experiment*, 2009(02):P02066, 2009. doi: 10.1088/1742-5468/2009/02/P02066.

[80] N E Huang, M-L C Wu, S R Long, S S.P Shen, Wendong Qu, Per Gloersen, and K L Fan. A confidence limit for the empirical mode decomposition and Hilbert spectral analysis. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 459(2037):2317–2345, 2003. ISSN 1364-5021. doi: 10.1098/rspa.2003.1123.

[81] H. E. Hurst. Long-term storage capacity of reservoirs. *Transactions of the American Society of Civil Engineers*, 116:770–808, 1951.

[82] E A F Ihlen. Introduction to multifractal detrended fluctuation analysis in matlab. *Frontiers in physiology*, 3, 2012.

[83] E A F Ihlen. Multifractal analyses of response time series: A comparative study. *Behavior research methods*, 45(4):928–945, 2013.

[84] Mads Ipsen and Alexander Mikhailov. Evolutionary reconstruction of networks. *Physical Review E*, 66(4):046109, Oct. 2002. doi: 10.1103/PhysRevE.66.046109.

[85] H. Jaeger. The echo state approach to analysing and training recurrent neural networks. Technical report, Technical Report GMD Report 148, German National Research Center for Information Technology, 2001.

[86] Herbert Jaeger and Harald Haas. Harnessing nonlinearity: Predicting chaotic systems and saving energy in wireless communication. *Science*, 304(5667):78–80, 2004. ISSN 0036-8075. doi: 10.1126/science.1091277.

[87] Sarika Jalan. Spectral analysis of deformed random networks. *Physical Review E*, 80(4): 046101, 2009. doi: 10.1103/PhysRevE.80.046101.

[88] Sarika Jalan and J N Bandyopadhyay. Random matrix analysis of network Laplacians. *Physica A: Statistical Mechanics and its Applications*, 387(2):667–674, 2008. doi: 10.1016/j.physa.2007.09.026.

[89] Biman Jana, Faruck Morcos, and J N. Onuchic. From structure to function: the convergence of structure based models and co-evolutionary information. *Physical Chemistry Chemical Physics*, 16:6496–6507, 2014. doi: 10.1039/C3CP55275F.

[90] Hawoong Jeong, Bálint Tombor, Réka Albert, Z N Oltvai, and A-L Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.

[91] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.

[92] Marc Kac. Can one hear the shape of a drum? *The American Mathematical Monthly*, 73(4): 1–23, 1966.

[93] Hetunandan Kamisetty, Sergey Ovchinnikov, and David Baker. Assessing the utility of coevolution-based residue–residue contact predictions in a sequence-and structure-rich era. *Proceedings of the National Academy of Sciences*, 110(39):15674–15679, 2013. doi: 10.1073/pnas.1314045110.

[94] J W Kantelhardt, S A Zschiegner, Eva Koscielny-Bunde, Shlomo Havlin, Armin Bunde, and H E Stanley. Multifractal detrended fluctuation analysis of nonstationary time series. *Physica A: Statistical Mechanics and its Applications*, 316(1):87–114, 2002.

[95] Kyle Kloster and D F Gleich. Heat kernel based community detection. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1386–1395, New York, NY, USA, 2014. ACM. doi: 10.1145/2623330.2623706.

[96] R. I. Kondor and John Lafferty. Diffusion kernels on graphs and other discrete structures. In *In Proceedings of the ICML*, pages 315–322, 2002.

[97] Reimer Kühn and Jort Van Mourik. Spectra of modular and small-world matrices. *Journal of Physics A: Mathematical and Theoretical*, 44(16):165205, 2011. doi: 10.1088/1751-8113/44/16/165205.

[98] Jarosław Kwapień and Stanisław Drożdż. Physical approach to complex systems. *Physics Reports*, 515(3):115–226, 2012.

[99] R B Laughlin, David Pines, Joerg Schmalian, B P Stojković, and Peter Wolynes. The middle way. *Proceedings of the National Academy of Sciences*, 97(1):32–37, 2000.

[100] D. M. Leitner. Energy Flow in Proteins. *Annual Review of Physical Chemistry*, 59(1):233–259, 2008. doi: 10.1146/annurev.physchem.59.032607.093606. PMID: 18393676.

[101] Anders Lervik, Fernando Bresme, Signe Kjelstrup, Dick Bedeaux, and J M Rubi. Heat transfer in protein–water interfaces. *Physical Chemistry Chemical Physics*, 12(7):1610–1617, 2010.

[102] Guifeng Li, Donny Magana, and R B Dyer. Anisotropic energy flow and allosteric ligand binding in albumin. *Nature communications*, 5, 2014.

[103] L. Livi, A. Giuliani, and A. Rizzi. Toward a Multilevel Representation of Protein Molecules: Comparative Approaches to the Aggregation/Folding Propensity Problem. *ArXiv preprint arXiv:1407.7559*, Jul 2014.

[104] L. Livi, A. Giuliani, and A. Sadeghian. Characterization of graphs for protein structure modeling and recognition of solubility. *arXiv preprint arXiv:1407.8033*, Jul 2014.

[105] L. Livi, E. Maiorino, A. Pinna, A. Sadeghian, A. Rizzi, and A. Giuliani. Analysis of heat kernel highlights the strongly modular and heat-preserving structure of proteins. *ArXiv preprint arXiv:1409.1819*, Sep 2014.

[106] L. Livi, E. Maiorino, A. Giuliani, A. Rizzi, and A. Sadeghian. A generative model for protein contact networks. *Journal of Biomolecular Structure and Dynamics*, 2015. doi: 10.1080/07391102.2015.1077736.

[107] L. Livi, E. Maiorino, A. Pinna, A. Sadeghian, A. Rizzi, and A. Giuliani. Analysis of heat kernel highlights the strongly modular and heat-preserving structure of proteins. *Physica A: Statistical Mechanics and its Applications*, 441:199–214, 2016. ISSN 0378-4371. doi: 10.1016/j.physa.2015.08.059.

[108] Lorenzo Livi and Antonello Rizzi. Graph ambiguity. *Fuzzy Sets and Systems*, 221:24–47, 2013. ISSN 0165-0114. doi: 10.1016/j.fss.2013.01.001.

[109] Lorenzo Livi, Enrico Maiorino, Antonello Rizzi, and A. Sadeghian. On the long-term correlations and multifractal properties of electric arc furnace time series. *International Journal of Bifurcation and Chaos*, 26(1):1650007, 2016. doi: 10.1142/S0218127416500073.

[110] J T-H Lo. Synthetic approach to optimal filtering. *IEEE Transactions on Neural Networks*, 5 (5):803–811, Sep. 1994. doi: 10.1109/72.317731.

[111] Renaud Lopes and Nacim Betrouni. Fractal and multifractal analysis: a review. *Medical image analysis*, 13(4):634–649, 2009.

[112] Mantas Lukoševičius and Herbert Jaeger. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, 2009. doi: 10.1016/j.cosrev.2009.03.005.

[113] N G Makarenko, L M Karimova, B V Kozelov, and M M Novak. Multifractal analysis based on the choquet capacity: Application to solar magnetograms. *Physica A: Statistical Mechanics and its Applications*, 391(18):4290–4301, 2012.

[114] D S Marks, L J Colwell, Robert Sheridan, T A Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3D structure computed from evolutionary sequence variation. *PLoS ONE*, 6(12):e28766, 2011. doi: 10.1371/journal.pone.0028766.

[115] J. Mazumdar and R.G. Harley. Utilization of echo state networks for differentiating source and nonlinear load harmonics in the utility network. *Power Electronics, IEEE Transactions on*, 23(6):2738–2745, Nov 2008. ISSN 0885-8993. doi: 10.1109/TPEL.2008.2005097.

[116] Patrick McDonald and Robert Meyers. Diffusions on graphs, poisson problems and spectral geometry. *Transactions of the American Mathematical Society*, 354(12):5111–5136, 2002.

[117] P. N. McGraw and Michael Menzinger. Laplacian spectra as a diagnostic tool for network structure and dynamics. *Physical Review E*, 77:031102, Mar. 2008. doi: 10.1103/PhysRevE.77.031102.

[118] M L Mehta. *Random Matrices*, volume 142. Elsevier/Academic Press, Amsterdam, 2004.

[119] J A Méndez-Bermúdez, A Alcazar-López, A J Martínez-Mendoza, F A Rodrigues, and T K D M Peron. Universality in the spectral and eigenfunction properties of random networks. *Physical Review E*, 91(3):032122, 2015. doi: 10.1103/PhysRevE.91.032122.

[120] Russell Merris. Laplacian matrices of graphs: a survey. *Linear Algebra and its Applications*, 10010:143–176, 1994. doi: 10.1016/0024-3795(94)90486-3.

[121] Rama Mishra and Shantha Bhushan. Knot theory in understanding proteins. *Journal of Mathematical Biology*, 65(6-7):1187–1213, 2012. doi: 10.1007/s00285-011-0488-3.

[122] Marija Mitrović and Bosiljka Tadić. Spectral and dynamical properties in classes of sparse networks with mesoscopic inhomogeneities. *Physical Review E*, 80(2):026123, Aug. 2009. ISSN 1539-3755. doi: 10.1103/PhysRevE.80.026123.

[123] Marija Mitrović and Bosiljka Tadić. Spectral and dynamical properties in classes of sparse networks with mesoscopic inhomogeneities. *Physical Review E*, 80(2):026123, 2009.

[124] Bojan Mohar. Laplace eigenvalues of graphs—a survey. *Discrete Mathematics*, 09:171–183, 1992. doi: 10.1016/0012-365X(92)90288-Q.

[125] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, D S Marks, Chris Sander, Riccardo Zecchina, J N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011. doi: 10.1073/pnas.1111471108.

[126] Faruck Morcos, Biman Jana, Terence Hwa, and J N Onuchic. Coevolutionary signals across protein lineages help capture multiple protein conformations. *Proceedings of the National Academy of Sciences*, 110(51):20533–20538, 2013. doi: 10.1073/pnas.1315625110.

[127] Hidetoshi Morita and Mitsunori Takano. Residue network in protein native structure belongs to the universality class of a three-dimensional critical percolation cluster. *Physical Review E*, 79:020901, Feb 2009. doi: 10.1103/PhysRevE.79.020901.

[128] M S Movahed and Evalds Hermanis. Fractal analysis of river flow fluctuations. *Physica A: Statistical Mechanics and its Applications*, 387(4):915–932, 2008. doi: 10.1016/j.physa.2007.10.007.

[129] Radhakrishnan Nagarajan. Reliable scaling exponent estimation of long-range correlated noise in the presence of random spikes. *Physica A: Statistical Mechanics and its Applications*, 366:1–17, 2006. doi: 10.1016/j.physa.2005.10.020.

[130] Thomas Neusius, Isabella Daidone, I M Sokolov, and J C Smith. Subdiffusion in peptides originates from the fractal-like structure of configuration space. *Physical Review Letters*, 100(18):188103, 2008.

[131] M E J Newman. Assortative mixing in networks. *Physical Review letters*, 89(20):208701, 2002.

[132] M. E. J. Newman. The structure and function of complex networks. *SIAM review*, 45(2): 167–256, 2003.

[133] M E J Newman. Modularity and community structure in networks. *Proceedings of the National Academy of Sciences*, 103(23):8577–8582, 2006.

[134] Vincenzo Nicosia, Manlio De Domenico, and Vito Latora. Characteristic exponents of complex networks. *EPL (Europhysics Letters)*, 106(5):58005, 2014.

[135] T. Niwa, B.-W. Ying, K. Saito, W. Jin, S. Takada, T. Ueda, and H. Taguchi. Bimodal protein solubility distribution revealed by an aggregation analysis of the entire ensemble of Escherichia coli proteins. *Proceedings of the National Academy of Sciences*, 106(11):4201–4206, 2009. doi: 10.1073/pnas.0811922106.

[136] Paweł Oświęcimka, Jarosław Kwapień, and Stanisław Drożdż. Wavelet versus detrended fluctuation analysis of multifractal structures. *Physical Review E*, 74:016103, Jul 2006. doi: 10.1103/PhysRevE.74.016103.

[137] Paola Paci, Luisa Di Paola, Daniele Santoni, Micol De Ruvo, and Alessandro Giuliani. Structural and functional analysis of hemoglobin and serum albumin through protein long-range interaction networks. *Current Proteomics*, 9(3):160–166, 2012. doi: 10.2174/157016412803251815.

[138] Tiago P. Peixoto. Eigenvalue spectra of modular networks. *Physical Review Letters*, 111(9): 098701, Aug. 2013. ISSN 0031-9007. doi: 10.1103/PhysRevLett.111.098701.

[139] C-K Peng, Shlomo Havlin, H E Stanley, and A L Goldberger. Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 5(1):82–87, 1995.

[140] R C Penner, Michael Knudsen, Carsten Wiuf, and J E Andersen. Fatgraph models of proteins. *Communications on Pure and Applied Mathematics*, 63(10):1249–1297, 2010. doi: 10.1002/cpa.20340.

[141] R C Penner, Michael Knudsen, Carsten Wiuf, and J E Andersen. An algebro-topological description of protein domain structure. *PLoS one*, 6(5):e19670, 2011. doi: 10.1371/journal.pone.0019670.

[142] Andrea Pinna, Nicola Soranzo, and Alberto De La Fuente. From knockouts to networks: establishing direct cause-effect relationships through graph analysis. *PloS one*, 5(10): e12912, 2010.

[143] Andrea Pinna, Nicola Soranzo, Ina Hoeschele, and Alberto de la Fuente. Simulating systems genetics data with sysgensim. *Bioinformatics*, 27(17):2459–2462, 2011.

[144] X-Y Qian, G-F Gu, and W-X Zhou. Modified detrended fluctuation analysis based on empirical mode decomposition for the characterization of anti-persistent processes. *Physica A: Statistical Mechanics and its Applications*, 390(23):4388–4395, 2011. doi: 10.1016/j.physa.2011.07.008.

[145] Shlomi Reuveni, Rony Granek, and Joseph Klafter. Anomalies in the vibrational dynamics of proteins are a consequence of fractal-like structure. *Proceedings of the National Academy of Sciences*, 107(31):13696–13700, 2010.

[146] Shlomi Reuveni, Joseph Klafter, and Rony Granek. Dynamic structure factor of vibrating fractals: Proteins as a case study. *Physical Review E*, 85(1):011906, 2012.

[147] Pratha Sah, L O Singh, Aaron Clauset, and Shweta Bansal. Exploring community structure in biological networks with random graphs. *BMC Bioinformatics*, 15(1), 2014. doi: 10.1186/1471-2105-15-220.

[148] MÁ Sánchez, J E Trinidad, J García, and M Fernández. The effect of the underlying distribution in Hurst exponent estimation. *PloS ONE*, 10(5):e0127824–e0127824, 2014. doi: 10.1371/journal.pone.0127824.

[149] A K Sangha and T Keyes. Proteins fold by subdiffusion of the order parameter. *The Journal of Physical Chemistry B*, 113(48):15886–15894, 2009.

[150] S. Scardapane, G. Nocco, D. Comminiello, M. Scarpiniti, and A. Uncini. An effective criterion for pruning reservoir's connections in Echo State Networks. In *Proceedings of the IEEE International Joint Conference on Neural Networks*, pages 1205–1212, Beijing, China, Jul. 2014.

[151] Francois Schmitt, Daniel Schertzer, and Shaun Lovejoy. Multifractal analysis of foreign exchange data. *Applied stochastic models and data analysis*, 15(1):29–53, 1999.

[152] M R Segal. A novel topology for representing protein folds. *Protein Science*, 18(4):686–693, 2009. doi: 10.1002/pro.90.

[153] Francesco Serinaldi. Use and misuse of some hurst parameter estimators applied to stationary and non-stationary financial time series. *Physica A: Statistical Mechanics and its Applications*, 389(14):2770–2781, 2010.

[154] Mark D Skowronski and John G Harris. Automatic speech recognition using a predictive echo state network classifier. *Neural networks*, 20(3):414–423, 2007.

[155] Marcin J Skwark, Daniele Raimondi, Mirco Michel, and Arne Elofsson. Improved contact predictions using the recognition of protein like contact patterns. *PLoS Computational Biology*, 10(11):e1003889, 2014. doi: 10.1371/journal.pcbi.1003889.

[156] Chaoming Song, Shlomo Havlin, and Hernán A Makse. Origins of fractality in the growth of complex networks. *Nature Physics*, 2(4):275–281, 2006.

[157] M P H Stumpf, Thomas Thorne, Eric de Silva, Ronald Stewart, H J An, Michael Lappe, and Carsten Wiuf. Estimating the size of the human interactome. *Proceedings of the National Academy of Sciences*, 105(19):6959–6964, 2008.

[158] Luciano Telesca and Vincenzo Lapenna. Measuring multifractality in seismic sequences. *Tectonophysics*, 423(1):115–123, 2006.

[159] Luciano Telesca, Vincenzo Lapenna, and Maria Macchiato. Multifractal fluctuations in seismic interspike series. *Physica A: Statistical Mechanics and its Applications*, 354:629–640, 2005.

[160] A. Topchy, A. K. Jain, and W. Punch. Clustering ensembles: models of consensus and weak partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(12): 1866–1881, October 2005. doi: 10.1109/TPAMI.2005.237.

[161] Constantino Tsallis. I. nonextensive statistical mechanics and thermodynamics: Historical background and present status. In *Nonextensive statistical mechanics and its applications*, pages 3–98. Springer, 2001.

[162] Berwin A Turlach et al. *Bandwidth selection in kernel density estimation: A review*. Université catholique de Louvain, 1993.

[163] Thomas W Valente, Kathryn Coronges, Cynthia Lakon, and Elizabeth Costenbader. How correlated are network centrality measures? *Connections (Toronto, Ont.)*, 28(1):16, 2008.

[164] E R van Dam and W H Haemers. Developments on spectral characterizations of graphs. *Discrete Mathematics*, 309(3):576–586, 2009. doi: 10.1016/j.disc.2008.08.019.

[165] Edwin R. van Dam and Willem H. Haemers. Which graphs are determined by their spectrum? *Linear Algebra and its Applications*, 373:241–272, Nov. 2003. ISSN 00243795. doi: 10.1016/S0024-3795(03)00483-X.

[166] Marco Vassura, Luciano Margara, Pietro Di Lena, Filippo Medri, Piero Fariselli, and Rita Casadio. Reconstruction of 3D structures from protein contact maps. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5(3):357–367, 2008. doi: 10.1109/TCBB.2008.27.

[167] Michele Vendruscolo, Emanuele Paci, C M Dobson, and Martin Karplus. Three key residues form a critical contact network in a protein folding transition state. *Nature*, 409 (6820):641–645, 2001. doi: 10.1038/35054591.

[168] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393, June 1998.

[169] Charles L. Webber, Alessandro Giuliani, Joseph P. Zbilut, and Alfredo Colosimo. Elucidating protein secondary structures using alpha-carbon recurrence quantifications. *Proteins: Structure, Function, and Bioinformatics*, 44(3):292–303, 2001. ISSN 1097-0134. doi: 10.1002/prot.1094.

[170] P G Wolynes. Evolution, energy landscapes and the paradoxes of protein folding. *Biochimie*, 30:50–56, 2014. ISSN 0300-9084. doi: 10.1016/j.biochi.2014.12.007.

[171] Zhaohua Wu, N E Huang, S R Long, and Chung-Kang Peng. On the trend, detrending, and variability of nonlinear and nonstationary time series. *Proceedings of the National Academy of Sciences*, 104(38):14889–14894, 2007. doi: 10.1073/pnas.0701020104.

[172] Bai Xiao, E. R. Hancock, and R. C. Wilson. Graph Characteristics from the Heat Kernel Trace. *Pattern Recognition*, 42(11):2589–2606, November 2009. ISSN 0031-3203. doi: 10.1016/j.patcog.2008.12.029.

[173] Wenying Yan, Jianhong Zhou, Maomin Sun, Jiajia Chen, Guang Hu, and Bairong Shen. The construction of an amino acid network for understanding protein structure and function. *Amino Acids*, 46(6):1419–1439, 2014.

[174] Xin Yu and D M Leitner. Anomalous diffusion of vibrational energy in proteins. *The Journal of Chemical Physics*, 119(23):12673–12679, 2003.

[175] Choujun Zhan, Guanrong Chen, and L F Yeung. On the distributions of laplacian eigenvalues versus node degrees in complex networks. *Physica A: Statistical Mechanics and its Applications*, 389(8):1779–1788, 2010. doi: 10.1016/j.physa.2009.12.005.

[176] Xiao Zhang, R R Nadakuditi, and M E J Newman. Spectra of random graphs with community structure and arbitrary degrees. *Physical Review E*, 89(4):042816, 2014. doi: 10.1103/PhysRevE.89.042816.

[177] Z-Z Zhang, S-G Zhou, and Tao Zou. Self-similarity, small-world, scale-free scaling, disassortativity, and robustness in hierarchical lattices. *The European Physical Journal B-Condensed Matter and Complex Systems*, 56(3):259–271, 2007.

[178] Zhongzhi Zhang, Zhengyi Hu, Yibin Sheng, and Guanrong Chen. Exact eigenvalue spectrum of a class of fractal scale-free networks. *EPL (Europhysics Letters)*, 99(1):10007, 2012. doi: 10.1209/0295-5075/99/10007.

[179] Todd Zorick and M A Mandelkern. Multifractal detrended fluctuation analysis of human EEG: Preliminary investigation and comparison with the wavelet transform modulus maxima technique. *PloS one*, 8(7):e68360, 2013.